Escuela Politécnica Superior
Departamento de Ingeniería Informática

An analysis of popularity biases in recommender system evaluation and algorithms

Dissertation written by

Rocío Cañamares Pérez

Under the supervision of

Pablo Castells Azpilicueta

Madrid, June 2019

Abstract

Recommendation technologies have become increasingly commonplace in everyday applications for the general public. Recommender systems make individualized suggestions of products or choices that users would probably find interesting or useful. Implicit in the concept of recommendation is the idea that each user may draw further benefit from a recommendation that is tailored to her personal tastes, as it seems is reasonable to expect that personalized algorithms should be the most effective, be it just because they consider a larger output space than a one-size-fits-all recommendation. It has been recently found however that non-personalized majority-based recommendations are not as suboptimal as one might expect. A strong bias towards popular items has been furthermore found in the top-performing personalized algorithms. Therefore, it would be relevant to understand to what extent, and under what circumstances, popularity is really an effective signal when recommending, and whether its apparent effectiveness is due to, as seems likely, a bias in the current offline evaluation methodologies.

This thesis addresses this question at a formal level, by identifying the factors that can affect the answer and modelling them in terms of dependencies between key random variables involving item rating, discovery and relevance. We find concrete conditions that guarantee popularity to be effective or quite the opposite, and settle the conditions under which there is a possibility of disagreement between observed and true accuracy. The clearest conclusions were reached for prototypical cases involving independence assumptions, without which we explain that any outcome is possible. Seeking further understanding of the general assumption-free case, we also study a particular case where item discovery is mainly a consequence of word-of-mouth in a social network. In addition, we provide a formal explanation of the bias towards recommending popular items that collaborative filtering methods present. We do so by developing a full probabilistic formalization of the k nearest neighbours scheme, upon which we also evidence the fundamental condition that makes this algorithm a personalized method and distinguishes it from pure popularity-based recommendations.

Resumen

Las tecnologías de recomendación han ido progresivamente extendiendo su presencia en las aplicaciones y servicios de uso diario. Los sistemas de recomendación buscan realizar sugerencias individualizadas de productos u opciones que los usuarios puedan encontrar interesantes o útiles. Implícita en el concepto de recomendación está la idea de que las sugerencias más satisfactorias para cada usuario son aquellas que tienen en cuenta sus gustos particulares, por lo que cabría esperar que los algoritmos de recomendación más eficaces sean los más personalizados. Sin embargo, se ha observado recientemente que recomendar simplemente los productos más populares no resulta una estrategia mucho peor que los mejores y más sofisticados algoritmos personalizados, y más aún, que estos tienden a sesgar sus recomendaciones hacia opciones mayoritarias. Por todo ello, es relevante entender en qué medida y bajo qué circunstancias es la popularidad una señal realmente efectiva a la hora de recomendar, y si su aparente efectividad se debe a la existencia de ciertos sesgos en las metodologías de evaluación offline actuales, como todo parece indicar, o no.

En esta tesis abordamos esta cuestión desde un punto de vista plenamente formal, identificando los factores que pueden determinar la respuesta y modelizándolos en términos de dependencias probabilísticas entre variables aleatorias, tales como la votación, el descubrimiento y la relevancia. De esta forma, caracterizamos situaciones concretas que garantizan que la popularidad sea efectiva o que no lo sea, y establecemos las condiciones bajo las cuales pueden existir contradicciones entre el acierto observado y el real. Las principales conclusiones hacen referencia a escenarios simplificados prototípicos, más allá de los cuales el análisis formal concluye que cualquier resultado es posible. Para profundizar en el escenario general sin suposiciones tan simplificadas, estudiamos un caso particular donde el descubrimiento de ítems es consecuencia de la interacción entre usuarios en una red social.

Además, en esta tesis proporcionamos una explicación formal del sesgo de popularidad que presentan los algoritmos de filtrado colaborativo. Para ello, desarrollamos una versión probabilística del algoritmo de vecinos próximos kNN. Dicha versión evidencia además la condición fundamental que hace que kNN produzca recomendaciones personalizadas y se diferencie de la popularidad pura.

Acknowledgments

The are many the people who have supported me in the development of this thesis, and I would like to dedicate the following lines to them.

First, I would like to thank my supervisor Pablo Castells. He has helped me so much in these years that any summary that I can write will be unfair and incomplete. I will try to do it nevertheless. I think that the best way to sum it up is by saying that this thesis belongs to him as much as it does to me. Day by day he has been there guiding me, interpreting the results, proposing new lines to follow, getting his hands dirty with the theoretical and empirical analysis, and I could go on. The are many people who have helped me, but without Pablo this thesis would not have been even possible. Thank you very much Pablo, it has been an honor working with you and learning from you, I hope we continuing doing it for a long time :-) (Yes, I am going to use smileys because these are the acknowledgments of my thesis and the reader will have enough time to read formal stuff throughout the rest of the document, do not be concerned about it ;-)).

I would not like to forget the rest of the Information Retrieval Group that, together with Pablo, have given me the opportunity of working and learning with them, thank you Fernando Diéz, Ivan Cantador, Alejandro Bellogín, Saúl Vargas, Ignacio Fernández and Pablo Sánchez. Especially I would like to thank Sofia Marina Pepa and Javier Sanz-Cruado, who practically joined the group at the same time as I did, for the wonderful moments and talks that we have shared together, both inside and outside the lab.

I would also like to thank Alistair Moffat for his supervision over the internship I did in The University of Melbourne. Alistair got involved with me much more of what his position officially requires, helping me with the accommodation, introducing me to his family (thank you so much Thau Mee for caring me) and taking me to visit different places of Melbourne. It was a hard time at the beginning without anybody from family or friends, and Alistair took care of me almost like a father, making me feel at home. I would also like to thank the rest of PhD students that I met in Australia for embracing me, thank you Sachini, Sameera, Namrata, Nicholas and Partha, I could not think of better lab colleagues for my visit to Melbourne. And of course, many thanks Guille for your support, meeting you in the Dubai airport was a very wonderful coincidence.

I would also like to dedicate a special mention to Isabel Landivar, my friend, personal coach and English teacher ;-). She has achieved that every week I feel more comfortable working in English. But her work has not been limited to this: she has also supported me with a lot of personal and PhD issues, anticipating the dilemmas that I may encounter and helping me dealing with them.

También quería dedicar unas líneas a todos los amigos que, durante la carrera, el máster y/o el doctorado han estado ahí y han hecho que hoy sea la persona que soy (espero no dejarme a nadie). Gracias Mario, Gallego y Juan por esas conversaciones a la hora de comer con las que aprendí a pensar y razonar como nunca antes. Gracias Lara, mi eterna compañera de prácticas, por todos esos ratos invertidos entre práctica y práctica, bueno, y también por las prácticas en sí jaja. Gracias Julia y Cris, por acogerme en vuestro grupo y hacerme ser mejor persona. Gracias a Pencho, Rafa y a la gente de la asociación de teleco por los buenos ratos pasados durante el máster. Gracias Sara y Pablo por aceptarme en vuestras comidas en los últimos meses, y presentarme a vuestro grupo de amigos, me lo estoy pasando genial con vosotros y estoy aprendiendo mucho. Gracias Omar, por esas tardes tirando gimnasios (referencia Pokemon, lo siento) con las que distraernos de nuestras respectivas tesis y, sobre todo por presentarme a toda esa gente con la que jugar al Hanabi el puente de la Constitución (lo siento para el que no lo pille ;-)).

Gracias también a David, por todo su apoyo en tantas cosas, especialmente cuando estaba a miles de kilómetros de distancia y sólo nos conocíamos desde hace unos meses. Alguien podría pensar que te decido sólo una línea de estos agradecimientos. Pero el hecho es que tú eres más mi presente y mi futuro, que mi pasado (por suerte :-)).

Por último, pero no menos importante, quería agradecer el apoyo de mi familia, especialmente el de mis padres. Vosotros sois los principales culpables de que hoy esté escribiendo estas líneas, de que me haya volcado con mi formación académica. Ambos sois un ejemplo de superación que siempre ha sido un referente para mí. Os volcasteis en vuestros estudios a pesar de todas las adversidades, y eso siempre ha sido la mejor enseñanza que he podido recibir. Gracias además por cuidar de mí, alegraros por mis éxitos y estar ahí apoyándome en mis fracasos (sí, vale, llegados a este punto ya tengo que reconocer que he llorado para escribir estos agradecimientos ;-)).

A ellos y a todas las personas que de una forma u otra han hecho posible la realización de este trabajo, lo sepan o no, muchas gracias.

Rocío Cañamares

May 2019

Contents

List of figures

List of tables

Table 6.1. Neighborhood size k in the kNN configuration on each dataset ($k = \infty$ indicates all items are taken as neighbors)..... 107
Table 6.2. Smoothing parameter μ for each probabilistic kNN variant on each dataset. 107

Chapter 1

Introduction

1.1 Motivation

Since their beginning in the early 1990s, recommender systems have progressively extended their presence into many day-to-day technologies (Linden et al. 2003). Nowadays, they are familiar elements for users of applications, services and tools of the most common environments. Most people are used today to Youtube recommending videos related with their interests, to Spotify suggesting new music to listen to, to Twitter, Linkedin or Facebook recommending a contact to connect with, to Google Play suggesting applications for their smartphones, or to any online store (Amazon, Fnac, etc.) recommending products to people based on their previous interactions in the platform.

Conceptually, a recommender system aims to predict the interests of the user by observing her interactions with the system. Based on such predictions, the system suggests new options that the user may find useful or interesting. Implicit in this concept is the idea that user satisfaction can be enhanced by personalizing and tailoring recommendations to the individual tastes of the user. Research in the recommender systems field is, in a way, taking this as a given. However, one of the questions that motivate this thesis is: to what extent do we need this personalization, how much of it, and how much room for enhancement we have between a non-personalized and a personalized approach? We may intuit that the answers to these questions may not be simple and have several sides to them (as have the questions) – a complexity that is precisely the object of this thesis.

Effective non-personalized alternatives to a personalized recommendation algorithm typically consist of aggregated user opinions that reflect majority trends – so-called popularity (Cañamares and Castells 2018a). Typical majority signals are the count of people who have consumed an item, the count of people who have shown appreciation for an item, and the ratio between the two latter counts. Non-personalized popularity-based suggestions are in fact a widespread feature we can find in practically any application involving massive catalogues of choices. Services such as Amazon, Youtube, online newspapers, social networks, etc., have a section somewhere displaying the most popular options – the most watched, the most read, the most bought, etc. Moreover, academic research has found (Cremonesi et al. 2010) that the number of people that can be pleased by popular

choices can be in the same order of magnitude as the satisfaction a state-of-the-art personalized algorithm can achieve – in exchange for orders-of-magnitude smaller development and maintenance costs. In fact, popularity-based recommendations can be the best possible option in situations where the observed data is very sparse and does not provide enough information for personalized algorithms to produce an accurate recommendation for each user.

From a wider perspective, trying (and thus recommending) what most people like may not be optimal, but it seems at least a reasonable idea that can be useful in many cases. In fact, the adoption of the behaviour, opinions or findings of other people may benefit us from the experience and knowledge of others, to guide us in situations of uncertainty, and to reduce the cost of elaborating a decision from scratch (Bandura 1971, Meltzoff & Prinz 2002, Miller & Dollard 1979). Thus, there is much at which one person resembles other, and in many cases what is good for one is also good for the other. In other words, we have a lot in common with most of our peers.

Even when we decide to personalize, researchers have recently found that the most effective personalized algorithms (in particular collaborative filtering methods) are strongly biased to majority tastes in common datasets (Cremonesi et al. 2010). Popularity thus seems like a trend one cannot escape from if we aim to achieve effective recommendations. Worse yet, offline evaluation has been shown to display a strong bias towards favouring popular recommendations. This bias can cast doubt as to whether algorithms are being properly compared and the state of the art has been properly established.

All these issues motivate the research undertaken in this thesis towards better understanding the effect of popularity in the development, behaviour and evaluation of recommendation technology.

To sum up, the research proposed in this thesis addresses the following questions:

- Is popularity really an effective signal for producing accurate recommendations?
- Does the answer depend on the specific popularity variant we consider? For instance, would there be a difference between computing the popularity of a product as the number of people who like or consumed it vs. the ratio of consumers of the item who liked it?
- Can we identify fundamental conditions that may determine the answer to the above questions? For instance, does the popularity effectiveness depend on how users discover the items? Or could it depend on the user behaviour, namely, on whether they are more likely to manifest their positive preferences than their negative ones?
- How does this generalize to state of the art collaborative filtering algorithms?
- Regarding the comparison between two or more algorithms, could observed effectiveness contradict the real one by declaring as winner an algorithm that is not?

1.2 Research goals

Based on the context and the questions previously formulated, the general objective of this thesis is to study to what extent and under what circumstances a popularity-based recommendation is an effective technique or not. To make progress towards such objective, this work has the following specific research goals:

- **RG1.** Identify a reduced set of random variables that enable a sufficient description of the elements under analysis: popularity distributions, accuracy of recommendation, item discovery, user-item interactions, user appreciations, and system observations. Based on such random variables, formalize the distinction between true and observed recommendation accuracy and describe the optimal non-personalized rankings that maximize each of these two types of accuracy.
- **RG2.** Identify prototypical scenarios that can be described in terms of the identified random variables and their probabilistic dependencies, for which it is possible to prove a particular result with regards to the effectiveness of the different popularity variants.
- **RG3.** Check the theoretical results empirically.
- **RG4.** Formally understand the influence of popularity in collaborative filtering algorithms. As mentioned in the motivation, the bias in state of the art algorithms towards recommending popular options is known in the field. But the reasons of this trend have not been formally analysed and explained yet.

1.3 Contributions

The work developed in this thesis has resulted in several contributions, which we summarize next.

- A probabilistic framework that allows for the formal analysis of recommender systems' behaviour and effectiveness. The framework models the key elements that determine the effectiveness of different popularity variants as explicit random variables. The behaviour of popularity, and the congruence of offline evaluation with true effectiveness, can be stated as an issue of probabilistic dependencies and interactions between such variables.
- Along the way in this analysis, we state a revised ranking principle for recommendation, adapting the Probability Ranking Principle (PRP) of Information Retrieval (Robertson 1977) to the recommendation task assuming that items are recommended only to new potential consumers. The principle is derived from a formal expression for the (observed and true) expected effectiveness of a generic recommender system.

- The description of meaningful and interpretable situations as probabilistic dependency configurations. We verify in particular, theoretically and empirically, the occurrence of potential contradictions between observed and true metrics in certain configurations.
- New findings regarding the comparative effectiveness of different popularity variants. While the volume of interaction (rating count) achieves the best results in usual offline experiments, we find that the volume of positive interactions is a more reliable signal, and more importantly, the ratio of positive interactions (average rating) tends to actually produce the best results, when measured with unbiased samples of user tastes.
- Empirical proof of the influence that information diffusion phenomena may have in the observed popularity distribution and, thus, in the behaviour and performance of popularity-based recommendation. In particular, we see that extreme information diffusion levels can make popularity obtain even worse results than random recommendation.
- A probabilistic version of the k nearest-neighbour (kNN) algorithm. The formal reformulation of this classic algorithm is useful in itself (beyond its use in this thesis) in enabling different sorts of analysis and improvements (smoothing, adding new variables, etc.). In the present thesis, it has allowed us to verify the hypothesis upon which kNN relies for being effective, namely, the dependency between users' tastes.
- Formal proof of the connection between kNN and recommendation by popularity. The aforementioned probabilistic reformulation of kNN show that, in the absence of dependencies between users' tastes, kNN is reduced to popularity. In fact, different kNN variants are related to different popularity variants.
- A new dataset that contains user ratings for music. Due to its gathering process, this dataset allows us – and the community – to carry out the algorithm evaluation process in absence of external popularity biases, beyond those that reflect the real users' tastes. Moreover, it contains information about the discovery distribution, which in turn enables the recreation of standard evaluation experiments where extra relevance judgments are available to compute true accuracy and compare it with observed accuracy.

Most of the previous contributions are focused on the study of the popularity biases and their effects. Understanding such biases is a first step to find better means to cope with them and thus devise more reliable evaluation techniques (Castells and Cañamares 2018), a clear future line of this thesis.

1.4 Structure of the thesis

The thesis is structured as follows:

- Chapter 1 (Introduction) presents the motivation, research goals, contributions and publications related to this thesis.
- Chapter 2 (Popularity in recommendation) presents a preliminary empirical revision of the popularity effectiveness. First, we introduce what is meant by popularity and its different interpretations. Second, we compare its effectiveness with that of other representative state of the art algorithms. Third, we verify and illustrate the existence of biases towards popular products in such algorithms.
- Chapter 3 (Related work) reports and analyses the related work on the topics addressed by this thesis. We introduce a series of concepts related with recommender systems and their evaluation, and we group prior work according to the aspect of popularity they address.
- Chapter 4 (Popularity biases in recommender system evaluation) carries out a formal analysis of the effectiveness popularity, in its different variants, and in different situations. We propose a probabilistic framework upon which, subsequently, we express the expected (observed and true) effectiveness of a recommendation algorithm in terms of as few factors as possible, that allow to characterize different situations. By applying the previous model to the particular case of popularity, we study the factors on which its effectiveness depends. In addition, we deduce the optimal non-personalized criterion for observed and true accuracy.
- Chapter 5 (Popularity biases derived from social network dynamics) delves in one of the situations commented in Chapter 4, by simulating the discovery and interaction with items through communication in a social network, and observing how different aspects of the social interaction may affect the observations that can become available (as input) to a recommender system, and the resulting effectiveness of popularity.
- Chapter 6 (Popularity biases in the nearest neighbour's approach) studies the connection of the kNN collaborative filtering algorithm to popularity distributions. For that purpose, we develop a probabilistic reformulation of kNN that explicitly shows the connection with popularity, and allows us to express the main hypothesis that sustain the kNN algorithm.
- Chapter 7 (Conclusions and future work) summarizes the thesis and synthetizes the conclusions to which our research leads. We also introduce the potential research lines to follow as future work.
- Appendix A contains the translation into Spanish of Chapter 1.
- Appendix B contains the translation into Spanish of Chapter 7

1.5 Publications related with this thesis

The work carried out along this thesis has given rise to several publications in conferences and workshops of the Information Retrieval area. We list them, grouped by the chapter of the thesis they are related to:

Publications related to Chapter 4

The following three contributions are related with the formal analysis of the popularity effectiveness that we develop in Chapter 4.

- R. Cañamares and P. Castells. Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). Ann Arbor, Michigan, USA, July 2018, pp. 415-424.
  CORE A+ (full paper)
  Best paper award.
- R. Cañamares and P. Castells. From the PRP to the Low Prior Discovery Recall Principle for Recommender Systems. 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). Ann Arbor, Michigan, USA. July 2018, pp. 1081-1084.
  CORE A+ (short paper)
- R. Cañamares and P. Castells. On the Optimal Non-Personalized Recommendation: From the PRP to the Discovery False Negative Principle. Workshop on Axiomatic Thinking for Information Retrieval and Related Tasks (ATIR 2017) at the 40th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). Tokyo, Japan, August 2017.

Publications related to Chapter 5

The next publication addresses the generation of popularity biases in a social network and how such biases may affect the effectiveness of recommending by popularity.

- R. Cañamares and P. Castells. Exploring social network effects on popularity biases in recommender systems. 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014) at the 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, USA, October 2014.

Publications related to Chapter 6

In the following long paper of SIGIR 2017 we develop a probabilistic reformulation of the nearest neighbors recommendation approach. A reformulation that explicitly expresses the connection of this algorithm with popularity.

- R. Cañamares and P. Castells. A Probabilistic Reformulation of Memory-Based Collaborative Filtering – Implications on Popularity Biases. 40th Annual International

ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). Tokyo, Japan, August 2017, pp. 215-224.
CORE A+ (full paper)

Other publications related to the thesis

According to the idea of avoiding popularity biases, and any other kind of distortion that may mislead the conclusions of an offline evaluation experiment, in the following publication we propose an experimental methodology to obtain unbiased results.

- P. Castells and R. Cañamares. Characterization of Fair Experiments for Recommender System Evaluation – A Formal Analysis. Workshop on Offline Evaluation for Recommender Systems (REVEAL 2018) at the 12th ACM Conference on Recommender Systems (RecSys 2018). Vancouver, Canada, October 2018.

Chapter 2

Popularity in recommendation

We start setting the stage for the present work by defining some fundamental concepts and presenting basic facts involving recommendation and popularity, from which the research questions addressed in this thesis arise. We shall first recall the definition of the recommendation task, and settle the terminology to be used in the rest of this document. We shall likewise introduce precise definitions of popularity and its variants and will illustrate its effects in typical basic recommendation experiments for offline evaluation.

2.1 The recommendation task

Recommendation is often referred to as the complementary face of search (Belkin and Croft 1992), where the system takes the initiative to suggest the user new options or products without any explicit query from the user. The options subject to be recommended can be of many different types (news, services, products, events, persons, etc.) so the generic term "items" is commonly used to refer to them.

Recommender systems try to predict the users' preferences in order to anticipate which item could be of interest and recommend it. To do this, the algorithm operates with an observed sample (usually a small fraction) of these users' preferences, obtained from the interactions between the users and the system. There is a wide variety in type and nature of such interactions; playing a song in Spotify, watching a movie on Netflix, playing a video on YouTube, liking posts on Facebook or Twitter, etc.). The interactions can be taken as an implicit or explicit sign of positive or negative user preferences for the items. A common and useful simplification represents such interactions as a as numeric *rating* that reflect degrees of preference. Ratings can be further abstracted to a binary value indicating whether or not the user likes the item. We shall henceforth refer to user-item observations as "ratings" for short, regardless of the nature of the user-item interaction records. Let us denote with $r(u, i)$ the observed rating by a user u for an item i. We shall

generally assume, for simplicity, that ratings are binary, since this is enough for our purposes. When a rating reflects a positive preference, we shall say that the item is *relevant* for the user.

Given the context described above, where a set of ratings from users to items is available for the system, the recommendation task consists in sorting the non-rated items of a each user in descending order, according to the relevance that the algorithm estimates each item has for the user. We shall assume that repeated recommendations are not allowed, namely, items already rated by the users cannot be recommended. This is the typical case in most recommendation scenarios, where the value and usefulness of recommendation involves an (implicit or explicit) purpose discovery. Recommending consumed items is considered in some applications – but we do not address that task in our present research (Benson et al. 2016).

The basic task of a recommender system (the task most research in the field has focused on since its early years) consists in satisfying user preferences as accurately as possible. The accuracy was initially understood as a rating prediction error minimization problem, a view that has been more recently replaced by the goal to produce useful rankings where items the user likes are placed as highly as possible in the ranking

The large amount of algorithms developed for over two decades to solve the recommendation problem is widely known. We comment briefly the main strategies in Section 3.1. As opposed to the most sophisticated methods which seek to provide the best possible recommendation tailored to each user, there are alternatives which although simpler are still employed. One of the most common is the one we study in this thesis, recommending by reverse order or popularity. Before analysing the effectiveness of popularity as a recommendation criterion, we must precise clearly the possible definitions of such criterion, as we do in the next section.

2.2 Popularity in recommendation

The notion of popularity admits different interpretations with small differences of nuance. In general, the popularity of an item refers to the global perception of the item by part of a population of persons (or entities). This perception is defined by two elements, the volume of observed perceptions (number of users) and the sign (grade of positivity) of the perceptions. In the recommendation context, considering one or the other or both gives rise to three main possible definitions of the popularity of an item: **total popularity**, **relevant popularity** and **average rating**. We provide next a precise explanation of each of these variants.

Understanding the popularity of an item as the number of ratings it has is the simplest interpretation of popularity and possibly the most frequent in the recommender systems literature (Cremonesi et al. 2010). The distinction between relevant and non-relevant items allows to distinguish two variants in this type: *total* popularity and *relevant* popularity:

- **Total popularity**. We define total popularity as the total number of ratings of an item, regardless of whether they are positive or negative. In other words, this is the number of people that have been observed interacting with the item. This is the common definition of popularity that we usually find in literature (Cremonesi et al. 2010, Jannach et al. 2015).
- **Relevant popularity**. We define the relevant popularity of an item as its number of ratings indicating a positive preference, namely, the number of people who have given signs that they like the item.

An alternative of the previous popularity interpretations is considering the quotient of the two above notions:

- **Average rating**. We define the average rating of an item as the ratio of people who have been observed interacting with the item and have (implicitly or explicitly) evidenced a positive preference for the item. If the interaction records involve a numeric rating value, this can be generalized to the average rating value.

The reader may realize that the word "popularity" gets thus somewhat overloaded in our proposed terminology: it can refer to a general notion of majority taste (comprising the three above definitions, or taking an even more abstract broader sense), or it can more specifically refer to the two first precise definitions. We shall nonetheless take care along this document that the sense of the word is always clear from the context.

It is quite common that applications which gather user opinions about a product catalogue (Amazon, Google Play, IMDb, etc.) show the valuation of these products in terms of some of the previous popularity definitions. For instance, IMDb shows the average rating of each movie in a scale from 1 to 10 stars, together with the volume of observations it has (i.e. its total popularity). Curiously, in this platform the most rated movie (The Shawshank Redemption with 2,087,966 ratings) is also the movie with the highest average rating (9.3/10). Further specializations of our definitions can be considered such as, for instance, the amount of generated revenue (e.g. Avatar is the top grossing film of all times). The three variants defined above capture nonetheless a main fundamental general distinction, that is representative of other possible particularizations, and can be computed in any dataset.

We analyse in detail the comparison of the three variants of popularity in Chapter 4, but for this introductory chapter our primary focus will be (total and relevant) popularity, since it has been the most widely used.

2.3 Popularity effectiveness

The popularity of an item can be used as a ranking criterion to deliver non-personalized recommendations. One would expect however that this such simplistic approach would

	Nr. users	Nr. items	Nr. ratings
MovieLens	6,040	3,706	1,000,209
Netflix	480,189	17,770	100,480,507
Last.fm	992	176,892	904,309

Table 2.1. Volumetric details of MovieLens, Netflix and Last.fm datasets.

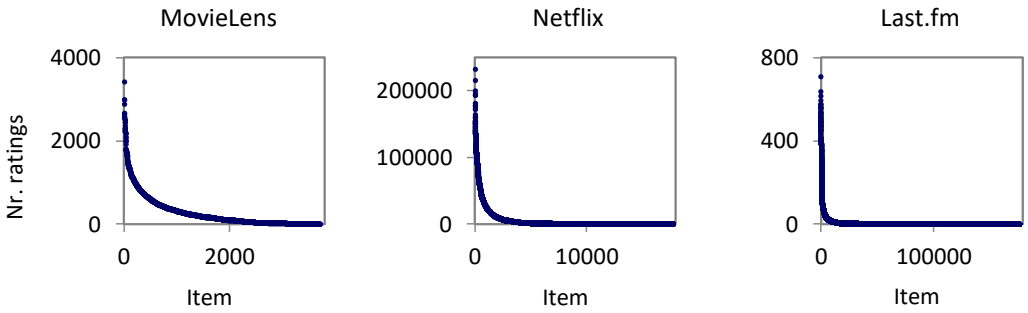

Figure 2.1. Rating distributions of MovieLens, Netflix and Last.fm.

produce considerably inferior results compared to top-of-the-line personalized algorithms. In order to observe how suboptimal popularity-based recommendations are, we set up an offline experiment much in a similar way to the typical offline evaluations reported in the recommender systems literature.

We take for this purpose three widely used public datasets: MovieLens, Netflix and Last.fm datasets, whose volumetric details are shown in Table 2.1. MovieLens is perhaps the most widely used dataset in the recommender systems research literature. It includes ratings for movies in a 1-5 scale by users of the MovieLens application (Harper & Konstan 2016). The Netflix dataset contains data of similar nature collected from Netflix subscriptors, and was released in 2006 in the Netflix Prize contest. The Last.fm dataset was collected by O. Celma (2010) and includes records of music tracks played by users on Last.fm. The recorded data for each play action includes the user ID, track, artist and timestamp. For our experiments we just aggregate this data into user / artist / playcount triplets and, as a simplification, we consider one or more playcounts as indicative of positive relevance. In MovieLens and Netflix we interpret ratings equal or higher than 4 as reflecting relevance, and lower values as non-relevant.

The number of ratings per item presents a strong biased distribution in all three datasets, as is common in the environments where recommender systems run. This can be clearly seen in Figure 2.1, where the rating distribution of MovieLens, Netflix and Last.fm is shown. The x axis represents the items sorted by decreasing order of total popularity and the y axis indicates the number of ratings of each item. We see that all three datasets display a typical long-tail popularity distribution where a few items are rated by many users while most of items receive a very few ratings.

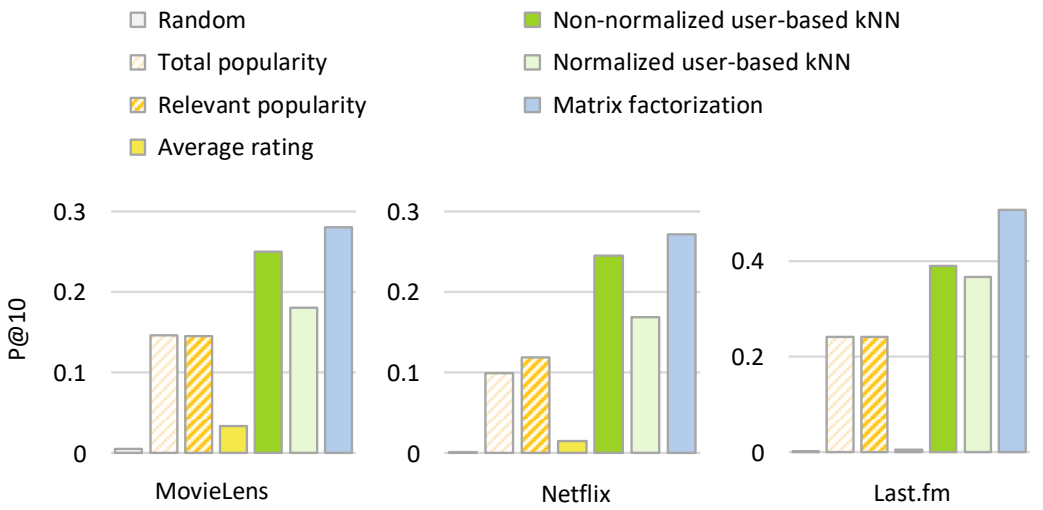

Figure 2.2. Results for common recommender algorithms on MovieLens, Netflix and Last.fm datasets.

To smooth the variance effects arising from the random split of the data, we average the results over a 5-fold cross-validation. As accuracy metric we select the precision in the top ten positions of the ranking (P@10) as a basic and representative metric for our current illustrative purposes.

Using this data, we test popularity-based recommendations along with a couple of representative personalized collaborative filtering methods: k nearest neighbors (kNN) and matrix factorization (Hu et al. 2008), as well as random recommendation for a sanity-check reference point. For the average rating, we require a minimum of 5 ratings for an item to be recommended. We take normalized and non-normalized user-based variants – with cosine similarity – as representative of the nearest-neighbour algorithm (Ning et al. 2015). For the normalized variant, we set a minimum number of 5 neighbour ratings for an item to be recommended to a target user. We select the best neighbourhood k for each kNN version (in terms of P@10) by grid search. Accordingly, kNN takes $k = 10$ neighbors on all datasets in the normalized variant, and $k = 100$ in the non-normalized one. The only exception is $k = 50$ in the non-normalized version on MovieLens.

Regarding the matrix factorization algorithm, we use the one proposed by Hu et al. (2008), since it is the most effective among those tested in recent years by our research group, and one of the fastest in execution time. We informally tune the parameter values based on previously reported configurations (Hu et al. 2008, Vargas & Castells 2014) and our own experience with well-behaving values for this algorithm. Finally, we take $k = 20$ factors, $\alpha = 1$, and $\lambda = 0.1$, with 20 iterations on all datasets, except $k = 50$ on Netflix.

Figure 2.2 shows the value of P@10 for the previous algorithms and the three popularity variants. Such results are in line with what other authors have observed in prior

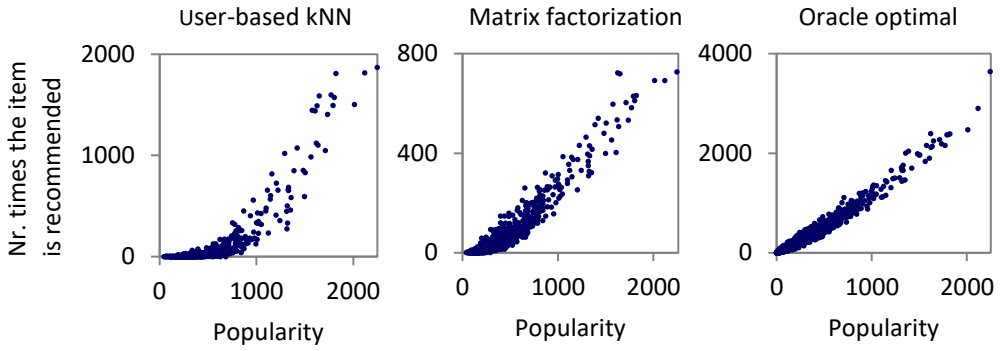

Figure 2.3. Popularity bias in different recommendation methods in the MovieLens dataset. Each point in the plots represents an item, the x axis shows the number of relevant ratings of the item, and the y coordinate is the number of times (i.e. the number of users to whom) the item is recommended in the top 10 by the corresponding algorithm.

work (Cremonesi et al. 2010, Jannach et al. 2015). We see that popularity (namely the "total" variant) achieves indeed lower accuracy than collaborative filtering, but its effectiveness is in the same order of magnitude (about half as effective) as the best personalized algorithm. This is not negligible for such a simple recommendation approach, requiring basically no scientific skills, lowest development and maintenance costs, and being capable to deliver reasonable recommendations even to new users with no records in the system.

As to the different popularity variants, we find that the total and relevant popularity achieve a very similar accuracy, with a tiny advantage for the latter. In contrast, the average rating achieves a very low precision. It should also be highlighted that the normalized variant of kNN is far below the non-normalized one. This is in line with the results reported by other authors but has not been explained yet. We will provide further explanation of this effect and the rest of the behaviours in Chapters 4 and 6.

2.4 Popularity biases in personalized recommendation

Having observed the non-negligible effectiveness of popularity alone, we now turn to analyse if there is some relation between the outputs of relevant popularity and state of the art collaborative filtering algorithms. We do so by measuring the number of times personalized algorithms recommend each item (say in the top 10 of the ranking) with the popularity of the item. Figure 2.3 show this as a scatterplot for the MovieLens dataset (on Netflix and Last.fm they present a similar behaviour).

We see that the best-performing personalized algorithms in Figure 2.2 – i.e. non-normalized user-based variant of kNN and matrix factorization – present a strong bias towards majority taste. In fact, their plots show that the number of times an item is recommended

by the algorithms is roughly proportional to its popularity. This is in line with what other authors have found in recent prior work (Marlin & Zemel 2009, Jannach et al. 2015).

But even more remarkably, we see in Figure 2.3 that the best possible recommendations (oracle recommendations) present even a stronger popularity bias. We build such optimal recommendations by randomly ranking all the items with positive test ratings by the target user at the top of the recommendation for such user. This would suggest that popularity is a trend one can definitely not escape from, since even the best possible recommendation (in terms of accuracy) involves an unequivocal popularity bias.

In the chapters that follow we will seek to explain why popularity is such an effective signal despite being an overly simplistic approach, and whether we may always expect this to be the case or we might run into exceptions. We will likewise seek some structural reason for the strong popularity bias in collaborative filtering. We will also wonder whether what we just observed is a final conclusion, or may be the result of some distortion in the evaluation methodology, the metrics, the data – or all of this together. Finally, given that we just observed that popularity is a trend within collaborative filtering, we shall wonder whether the findings on popularity translate to equivalent properties for the personalized algorithms that have popularity as an implicit component, and can help understand (and optimize) their behaviour.

Chapter 3

Related work

In this chapter, we review the main concepts and related work for the questions and goals addressed in this thesis. We briefly recall the broad types of recommendation algorithm, and the common approaches to evaluate them. We then focus on the prior work related, in one way or another to issues of popularity in recommendation.

3.1 Recommendation algorithms

Recommender systems have consolidated over the past three decades as a distinctive research area (Ricci et al. 2015), with a strong commercial impact and development (Linden et al. 2003). Research in the field focused for a long initial period on the development of algorithms that accurately predicted user preferences for items (Herlocker et al. 2004). More recently, algorithmic development has shifted towards targeting effective item rankings where users will find items they like early in the item lists (Cremonesi et al. 2010, Steck 2013). Currently, these perspectives have widened beyond oversimplified formulations, to consider other desirables qualities and concerns in recommendation such as novelty and diversity (Adamopoulos & Tuzhilin 2014, Celma & Herrera 2008), fairness (Dwork et al. 2012, Mehrotra et al. 2018) and business performance (Wu et al. 2017), aiming to address the complexities of real recommendation scenarios involving multiple stakeholders (Abdollahpouri et al. 2017) and a long-term relationship between users and the system (Li et al. 2016, Kawale et al. 2015).

Based on the input data and how the data is handled, it is common to distinguish of the following types of recommendation algorithms:

- Collaborative filtering. The methods of this group guess the target user's opinion on the items she does not know from the ratings of people with similar tastes. Recommendations are thus based on other users' opinions and the similarities that such

opinions present with those of the target user. The algorithms of k nearest neighbours and matrix factorization, cited in previous chapter, belong to this group. The latter usually performs slightly better than the former.

- Content based. The algorithms belonging to this group consider the characteristics of the items rated by the target user for estimating her preference for other items with similar characteristics. Therefore, the items do not need to be rated by any user in order to be considered as potential candidates. This allows content-based algorithms to recommend new items, or items with very few ratings, unlike collaborative filtering methods. However, the former do not take into account the opinions of other users.
- Recommendation algorithms based on social network. They use the preferences evidenced by explicit social contacts of the target user in the social network to predict her interests.

Collaborative filtering methods generally achieve higher accuracy than content-based recommendations. In fact, the top performing algorithms of the state of the art – such as implicit matrix factorization (Hu et al. 2008), SLIM (Ning & Karypis 2011), BPR (Rendle et al. 2009) – belong to the group of collaborative filtering methods.

It is not uncommon, however, to use trivial recommendations both in experiments and in commercial solutions, as popularity-based or random recommendations. We will pay considerable attention to the former in this thesis, but the latter is also quite used as sanity check or even as a way to gather unbiased user preferences (Gruson et al. 2019).

3.2 Evaluation

There are multiple ways of measuring the quality of a recommendation, depending on the characteristic we want to optimize (Herlocker et al. 2004, Shani & Gunawardana 2015). In this section we explain first the main types of metrics used to evaluate recommendations and then the methodologies followed to compute such metrics.

3.2.1 Dimensions and metrics

Among the dimensions employed nowadays for measuring the quality of a recommendation, we can distinguish two main types: accuracy, and novelty and diversity

Accuracy

Accuracy metrics evaluate the level of success of the recommendation. They are thus focused on the users' opinion about the recommended items, and can be grouped in error metrics and ranking metrics.

Due to historic heritage of the data mining field, error metrics have been for a long time the most used metrics for measuring the effectiveness of a recommender system.

They focus on the rating value predicted by the algorithm, and measure how close such value is to the real rating assigned by the user to the target item. Most common metrics of this group are Mean Absolute Error (MAE) – defined as the mean of the absolute differences between predicted and real ratings – and Root Mean Squared Error (RMSE) – equivalent to MAE but taking the square of the differences (to accentuate the greatest values) and the root square of the mean (Herlocker et al. 2004).

However, in the past years it has been accepted (Cremonesi et al. 2010, Steck 2013) that these metrics are not representative of the real user satisfaction. In other words, that the values of these metrics do not correlate with the probability that users accept recommendations. For this reason, the community has recently adopted the accuracy metrics characteristic of the Information Retrieval field, as precision, recall or nDCG (Baeza & Ribeiro 2011). These metrics measure the ranking quality and seem to represent better the final utility for real consumers than error prediction metrics. Moreover, these metrics open the door to the evaluation of ranking based algorithms – as popularity – that could not be measured in terms of rating error prediction. In this thesis we will precisely focus on ranking-based accuracy metrics as the ones we have just mentioned.

Novelty and diversity

By the beginning of the 2000's the community started to pay attention to other aspects), beyond accuracy, that may be desirables in a recommendation (Herlocker et al. 2004, Smyth & McClave 2001), as the novelty or diversity of the comprising items (McNee et al. 2006, Ziegler et al. 2005).

Novelty metrics reward the recommendation of items that the user probably does not know, whereas diversity metrics seek to potentiate the recommendations of not too similar items. There are many metrics belonging to each of these groups, but to name a few, long tail novelty (Celma & Herrera 2008) or unexpectedmess (Adamopoulos & Tuzhilin 2014) measure ranking novelty, while intra-list diversity (Zhang & Hurley 2008, Ziegler et al. 2005) or sales diversity (Fleder & Hosanagar 2009) are focused on the diversity of the recommendation (Castells et al. 2015).

Other dimensions

Accuracy, novelty and diversity metrics are complemented in practice by additional dimensions which aim to measure the business performance, as the number of purchases, the basket size, the click-through rate, etc.

3.2.2 Evaluation methodology and experimental design

Industry evaluations are based nowadays mainly on online A/B testing (Siroker and Koomen 2015), with evaluations carried out on the own production platforms. In this scope, the recommendation effectiveness is measured essentially in terms of the increase in the amount of interactions (purchases, clicks, play-counts, etc.) between the users and

the recommended elements, but also in terms of clients' fidelity or economic performance. Online evaluation is expensive, takes time and involves risks in decreasing the production system quality and user perception thereof. Moreover, it requires the availability of a working application with a significant user basis, which is not within reach for all researchers in the field. For these reasons, online evaluation is often complemented (or replaced by) with offline experiments.

Offline evaluation methodologies (both ranking and error oriented) divide the available data into a training set – given as input to the recommender systems – and a test set – that is used to evaluate the algorithm capacity to guess the users' tastes. In recent dates it has been observed how the way to carry out this separation (that is, essentially, a sample) between training and test can condition the evaluation outcome (Bellogín et al. 2017).

The gathering of the user interactions is, in essence, a sampling process bounded to biases whose nature and explanation has only started to be analysed (Gruson et al. 2019, Yang et al. 2018). For a long time, the recommender systems literature barely addressed the origin and characteristics of the data that algorithms take as input – and that is also used as ground truth for evaluation. Some work has analysed issues of reliability and quality of the data (Cheng & Hurley 2009); more recently, studies focused on analysing the popularity biases in the data, as we discuss in the next section.

3.3 Popularity on recommender systems

The studies related with the popularity issues can be grouped in four generic areas, depending on the aspect they address: verifying the existence and influence of popularity biases; proposing metrics which take such biases into account and try to mitigate their effects; reproducing and studying the processes that give rise to these biases – including those processes drawing from social phenomena; enhancing novelty in recommendation as the opposite of popularity. We briefly discuss work in these directions in the following subsections.

3.3.1 Existence and influence of popularity biases

The shift from error metrics (MAE, RMSE) to ranking metrics – precision, recall, nDCG, etc. – has enabled the evaluation of algorithms that, as popularity, cannot be evaluated with error metrics. A pioneering study in this line (Cremonesi et al. 2010) compared both ways of measuring and verified the results that we show in previous chapter. Namely, that when employing ranking metrics to evaluate, recommendation by pure popularity obtained a conspicuously high performance, compared with other more complex and personalized algorithms. A later study by Cremonesi et al. (2014) further showed that this performance decreases when most popular items are removed, from which it can be deduced that the popularity effectiveness relates to the bias of the popularity distribution

(i.e. the distribution that states how popular is each item). However, Cremonesi et al. did not delve into an analysis or finding an explanation for this connection between the recommendation performance and the data biases.

Bellogín et al. (2017) confirm this connection by using simulated ratings as input to recommendation algorithms (an also as ground truth to evaluate them). They increase the bias of the popularity simulated distribution and observe how this increase makes the popularity algorithm obtains better results. They also provide a formal intuition of this behavior, by explaining that under a random data split the number of ratings of an item in training and test is proportional. Thus, the higher the bias of the rating distribution, the larger the number of ratings in test of the most popular item, and then the higher the effectiveness obtained when recommending such item (He & Garcia 2009). In addition, Bellogin et al. also verify that the previous effects appear when employing the accuracy ranking metrics imported from Information Retrieval. This study is therefore a first work in the line of understanding why popularity biases affect popularity effectiveness, but it did not delve into explaining the causes of such biases or whether they are distorting the evaluation or not.

Such biases are not only observed in the uneven rating distribution (Little & Rubin 1987), but also in the fact that positive ratings are typically more frequent than negative ones. That is, the absence of ratings is not uniform, it depends on their value (Goel et al. 2010, Krishnan et al. 2014, Marlin et al. 2007, Pradel et al. 2012, Steck 2013). This missing not at random (MNAR) condition of the ratings was firstly verified by Marlin et al. (2007) when explicitly asking about it to real users. A large majority of such users expressed that their opinion about items considerably affects the decision to rate them.

The biased data distribution also has an important impact on personalized algorithms, and collaborative filtering methods in particular, which tend to recommend popular items. Our observations of Section 2.4 in this line agree with those of other authors who already observed this effect (Cremonesi et al. 2010, Jannach et al. 2015, Marlin & Zemel 2009). In particular, Jannach et al. (2015) verified and measured this correlation with popularity, and proposed approaches to counter it. However, the confirmation of this trend towards recommending popular items is empirical, and a formal and precise explanation of this phenomenon has not been proposed yet. We do it in Chapter 6, by formally proving the existence of a popularity component in the k nearest neighbors approach.

Previous studies point out the existence of popularity biases, and show that such biases can substantially influences both recommendations (they are biased towards popular items) and evaluation outcomes using ranking metrics (recommending popular items is rewarded). Some of them even propose solutions to mitigate this influence in metrics and algorithms, as we explain next. However, the characterization and explanation of the causes that give rise to different biases have not been addressed yet. Nor if such biases are desirable or not or to what extent they may lead to a distortion in the results of an

experiment (not only in the metric values but also in the comparison sign between algorithms). In Chapter 4 of this thesis we study precisely the variables that determine such biases, and present distinct situations – characterized for how such variables are related – in which we can check the existence or not of contradictions between the observed effectiveness (measured with the observed data) and the real one.

3.3.2 Popularity bias mitigation

Based on realisations as those we summarize in previous section, recent works have proposed different compensation mechanisms both in algorithms and metrics, that take into account the uneven distribution of data over the set of items (Steck 2010, Steck 2011, Zhao et al. 2013). These studies address the influence of popularity biases on collaborative filtering methods, as a problem we should solve, arguing that a recommendation of less popular items is more valuable.

Zhao et al. (2013) consider that rating something liked by few users provides more information about the user's tastes than a rating on a popular item. Consequently, they propose promoting less popular items in recommendations. In the same line, Steck (2010) defines target functions for recommendation that reward less popular items, and then employs such functions to both train and evaluate the algorithms.

The procedure followed by Steck (2010, 2011) is also relevant for this thesis for its distinction between observed and true metrics. The aim of Steck is to measure or approximate an estimate of the real recommendations' effectiveness, the one that we would obtain using the real and complete users' tastes, and not only the observed ones. However, faced with the impossibility of having full knowledge of user preferences, Steck proposes a series of metrics based on a main assumption: relevant ratings are uniformly distributed over items. This assumption allows him to assert that the metrics he proposes, when evaluating with observed data, provide unbiased estimates of the true metric values.

Bellogin et al. (2017) also propose methods to mitigate the effects of popularity biases, but they focus on the experimental methodology instead of on the metrics or the algorithms. Thus, they propose two new split protocols. The first one consists on partitioning the set of items by popularity strata, computing the accuracy for each stratum, and average the obtained values over all strata. This way the popularity bias in each item segment is smaller than in the full set of items, and so is its impact on accuracy. The second approach consists in building a test set where all items have the same amount of ratings.

On a closely related line, drawing from the areas of machine learning and statistics, the bias in offline evaluation has been addressed as an issue of mismatch between the data gathering policy (in this case, the free user interaction with the system) and item selection by the recommendation algorithms to be evaluated. Thus, techniques (such as inverse propensity scoring) have been explored to remove the biases in the evaluation (Gilotte et al. 2018, Gruson et al. 2019, Swaminathan et al. 2019, Yang et al. 2018) and the evaluated algorithms (Schnabel et al. 2016).

3.3.3 Popularity vs. novelty

One of the main limitations of recommending popular items is the lack of novelty (Celma & Herrera 2008, Castells et al. 2015, Lee & Lee 2011, Nakatsuji et al. 2010, Oh et al. 2011, Onuma et al. 2009). Popular items are familiar to a majority of users, even for many people who may not have made their opinion observable though a rating, and thus recommending these items may be too obvious and lack utility.

A wide strand of research in the recommender system fields has aimed to measure and enhance the novelty of recommendation (Adamopoulos & Tuzhilin 2014, Adomavicius & Kwon 2012, Celma & Herrera 2008), and one might think this is all we need to deal with popularity: simply avoid it. However, this is an incomplete answer to the questions about popularity in recommendation. While lack of novelty is an obvious drawback of popularity, the effect of popularity on pure accuracy should be properly understood. Even avoiding the head of the popularity distribution, some items are still more popular than others, and we need to understand the difference between recommending each of them.

3.3.4 Generation process of popularity

The popularity distribution over items is derived from the rating generation process, which leads some items to concentrate higher numbers of ratings than others. This process has been studied in prior work, mainly with the objective of formally modelling the user behaviour (Borghol et al. 2011, Harper et al. 2005), or predicting the popularity reached by different items (Hensinger et al. 2013, Ratkiewicz et al. 2010, Shen et al. 2014, Szabo & Huberman 2010, Zhang et al. 2014), rather than understanding how such biases might distort recommendations. The retroactive effect that recommender systems' intervention has on the popularity distribution has also be studied, as a particular source of item popularity (Adamopoulos et al. 2015, Fleder & Hosanagar 2009, Sharma et al. 2015, Sinha et al. 2016): if a recommender system is biased towards majority taste, then the most popular items are the most recommended, thus the most exposed to users and to chances of obtaining even more ratings. This effect also has been observed in social networks, such as Twitter, when the recommended items are the users themselves (Su et al. 2016).

The study of Salganik et al. (2006) is also worth being mentioned in this context. Such study reveals the presence of a certain degree of chance and uncertainty in the order in which products are discovered, even in absence of external influences. Salganik et al. (2006) further go on to analyse the role of social influence in popularity formation. The authors observed that when we inform users about other people's opinions, the popularity distribution becomes more skewed, at the same time that the uncertainty becomes higher as to what items will be the ones to become most popular. Other studies in this line confirm the effect of social influence in the increase of unpredictability (Abeliuk et al.

2017) and popularity biases (Wang & Wang 2014), and how this contributes to the fragility of the popularity signal (Bikhchandani et al. 1992).

Another social factor that may affect the rating generation process are the information diffusion phenomena (Newman 2010). A strong diffusion level, for instance, can make some items be discovered by a lot of users and thus increment the popularity of such items. In this line, Bakshy et al. (2012) and Doerr et al. (2012) study different characteristics of the social network that may boost information diffusion. In Chapter 5 we will see in more detail how this level of diffusion in a social network can affect the popularity distribution.

Chapter 4

Popularity biases in recommender system evaluation

As we set out in Chapter 2, the evaluation methodologies we consider in this thesis – the offline methodologies – divide the available data in a training set, given as input to recommender systems, and a test set which is hidden to the algorithms and used to evaluate their accuracy. The main concern of this approach is that only the relevance information contained in the test set is available to compute the accuracy metrics. Namely, the opinions that are not present in the test set are considered non-relevant, even though they may be in fact positive. A difference arises thus between the metric we can measure in the usual offline evaluation approach and the true metric value we would compute if we had full access to the missing relevance knowledge.

Alongside this difference is the fact that common datasets present strongly biased popularity distributions, as we illustrated in Chapter 2. This may lead us to suspect that the algorithms which consider such popularity signal in their recommendations might be rewarded somehow, especially since the best oracle recommendation is also biased towards popularity (see Figure 2.3 of Chapter 2). In the particular case of pure popularity-based recommendation, it is well-known that it can achieve suboptimal but non-negligible accuracy when evaluating with offline evaluation methodologies. The question is therefore open whether this apparent effectiveness is real or, on the contrary, it is a consequence of the previous methodological design along with the biased data.

In this chapter we address this question at a formal level, capturing in a probabilistic model the factors and conditions that affect the answer. The mathematical formulation allows us to study the effectiveness of popularity in different situations – expressed in terms of dependencies between random variables – and settle the conditions under which there is a possibility of disagreement between observed and true accuracy. A disagreement that we shall empirically verify with experimental results.

4.1 Probabilistic framework

In order to develop a formal analysis of the popularity effectiveness, we start by setting out the relevant concepts and variables involved in the recommendation task. These aspects allow us to define the mathematical framework upon which we are going to conduct the rest of the theoretical analysis. We shall take a probabilistic approach and describe these factors as random variables, in order to reason in terms of probabilities and expected values.

4.1.1 Random variables

The factors we handle to characterize distinct situations are those with some influence in the recommendation and evaluation outcomes. Thus, we explain them from the analysis of such tasks, for which we abstract ourselves from popularity and consider a generic recommendation algorithm.

Recommendation variables

As set out in Section 2.1, a recommender system is an algorithm that takes the observed interactions between users and items – positive and negative ratings in our study – as input and tries to predict the interest of the users in order to suggest them new products. The recommendations resulting from this process are therefore mainly dependent on the algorithm itself and the distribution of such input data. In our study, the objective is analysing the behaviour of a specific algorithm in different scenarios, so the variable that represents the algorithm is fixed. Variations between the recommendation outcomes come thus only from the data distribution, so we shall study its generation process in order to characterize distinct situations.

This data distribution is defined by two processes that can be described as stochastic: the process that drives the user to rate an item – which encompasses in turn many other subprocesses we do not need to consider for now, such as discovering the item, consuming it, deciding to rate it, etc. – and the personal taste of users with regards to items (relevance). As a typical simplification we shall consider relevance as a static and intrinsic variable between users and items, regardless whether the user knows the item or not.

We shall thus handle two random variables corresponding with these two kinds of relationships between users and items. Both variables are binary, and their domain is the sample space $\mathcal{U} \times \mathcal{I}$, where $\mathcal{U}$ is the set of users and $\mathcal{I}$ de set of items. First, we define the random variable $rated: \mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$ over the set of user-item pairs as $rated = 1$ if a rating by the user for the item is available in the dataset and 0 otherwise. Similarly, we define $rel: \mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$ as $rel = 1$ if the user likes the item (regardless of the presence or absence of rating), and 0 otherwise.

Throughout the thesis we will use the abbreviation $p(rated)$, $p(rel)$, etc., for $p(rated = 1), p(rel = 1)$, and so forth. Likewise, if all the random variables present in the expression have the same arguments, those will be placed in the condition. For instance, the probability $p(rated(u,i), rel(u,i))$ will be replaced for $p(rated, rel|u,i)$ for simplicity.

With these two variables and the dependencies between them we can thus describe in probabilistic terms most of the potential situations. For instance, an environment where positive feedback is more frequent that negative one can be described by the condition $p(rated|rel,i) > p(rated|\neg rel,i)$.

Our aim is therefore to express the criteria of the different popularity variants (total popularity, relevant popularity and average rating) as a function of the variables rel and $rated$, and then repeat the process with the effectiveness of a recommender system. After that, we will thus be able to compare both expressions and reason how effective popularity will be in different situations. In order to do that, we shall however consider first the variables that come into play in the evaluation process.

Evaluation variables

So far, in order to understand the factors that determine the behaviour of a recommendation algorithm we have ignored the way its effectiveness is going to be measured (metric, split protocol, etc). However, this can affect – even drastically – both the measurement results and indeed the recommendation itself, since it determines which part of the input data is supplied to the algorithm, as well as the inclusion or exclusion of some items in the ranking requested to the algorithm (Bellogín et al 2011, Said & Bellogín 2014).

This influence supports the definition of two new random binary variables, the variables $train$ and $test$. Both are defined on user-item pairs and take value 1 if $rated = 1$ and the rating was assigned to the training or test partition respectively, and 0 otherwise.

Our analysis shall also assume a random rating data split with a given ratio $\rho \in (0,1)$ of training data, independent from users and items (formally $p(train|rated,u,i) = \rho$). We consider a common data partition procedure which consists of iterating over each of the available ratings in the dataset, assigning it to the training or test set with probability ρ and $1 - \rho$ respectively.

4.1.2 Popularity rankings

We introduced in Section 2.2 three common interpretations of popularity, namely total popularity, relevant popularity and average rating. They give rise to three popularity-based recommendation algorithms whose ranking functions we denote by $pop(i), rpop(i)$ and $avg(i)$ respectively, for $i \in \mathcal{I}$.

Total popularity sorts the items according with the number of ratings they have in training ($pop(i) = |i_{train}|$), while relevant popularity only considers those ratings reflecting a positive preference ($rpop(i) = |i^{+}_{train}|$). Average rating, on its part, consists on the ratio of positive ratings ($avg(i) = rpop(i)/pop(i)$). Note that we are considering a binary version of average rating since it has a more tractable probabilistic formulation.

Using the random variables defined in the section 4.1.1 we can express these ranking criteria as follows: $pop(i) \propto |i_{train}|/|\mathcal{U}| = p(train|i)$, $rpop(i) \propto |i^{+}_{train}|/|\mathcal{U}| = p(train, rel|i)$, and $avg(i) = p(rel|train, i)$. Note that they are non-personalized recommendations and therefore the user is not fixed. Relevant popularity, for instance, ranks items by $p(rated, rel|i)$, but not specifically by $p(rated, rel|i, u)$ for each user.

Now we use that $p(train, \neg rated|i) = 0$ and that, according to the random split procedure described above, the probability for a rating to be sampled for training is independent from both the item and the rating value (and equal to the split ratio ρ). Therefore, we have that $p(train|i) = \rho\, p(rated|i)$ and $p(train, rel|i) = \rho\, p(rated, rel|i)$, and thus:

$$\begin{aligned} pop(i) &\propto p(rated|i) \\ rpop(i) &\propto p(rated, rel|i) \\ avg(i) &\sim p(rel|rated, i) \end{aligned} \tag{4.1}$$

We shall use the popularity ranking functions in this form in the rest of the thesis.

4.2 Expected precision

Given a recommendation for a user, its precision $P@k$ is defined as the number of relevant items in the top k positions of the ranking. We chose to model this metric since it is both representative of ranking-based accuracy metrics and tractable in probabilistic terms. For the same reason, we restrict our formal study to $P@1$. In the experiments of section 4.8 we will see that our analysis and results generalize well empirically to other accuracy metrics and common deeper cutoffs.

Taking $k = 1$ turns $P@k$ into a binary function that is equal to 1 if the top ranked item is relevant for the target user, and 0 if it is not. This makes it easier to reason about the expected value of this metric: as a binary function, the expectation of $P@1$ for a given recommendation R is the probability of taking value 1: $\mathbb{E}[P@1|R] = p(P@1 = 1|R)$.

As we point out at the beginning of this chapter, we shall distinguish between *observed precision* – computed in the typical evaluation experiments from the partial relevance information contained in the test set and that we denote by $\hat{P}$ – and *true precision* – the one we would obtain if we had full relevance knowledge and that we denoted by P. Accordingly, we have $P@1 = 1$ iff the target user likes the top-ranked item, whereas $\hat{P}@1 = 1$

iff the target user likes the top item *and* a rating by the user for the item is present in the test set. From these definitions it gets clear that $\hat{P} \leq P$, namely observed precision is a lower bound of true precision.

Now we need to take care about the possibility that the top ranked item has already be rated by the target user. As we introduce in Section 2.1, recommender systems are usually employed with a purpose of discovery, as a compliment of search engines. Consequently, they typically exclude from the recommendation those items the target user has already been observed interacting with. In terms of offline evaluation, this approach means that items with a training rating of the target user must be excluded from the ranking offered to this user.

This exclusion is commonly carried out by the external evaluation framework before invoking the algorithms, by removing such items from the list of potential candidates to be recommended. We shall however reason here as if the algorithms did score and rank all the items – including those with an observed interaction – whereupon the external system takes care of taking them out from the ranking before delivering the recommendations. This shall simplify our analysis without loss of equivalence to the usual procedure.

Item exclusion taking place after the recommendation process means that the recommended ranking R is made up of all the items. We shall however take such exclusion into account in the metric computation, namely, $P@1$ takes value 1 iff the first ranked *recommendable* item in R is relevant. Where recommendable means that the item has not a rating in the training set.

Let this first recommendable item be R_k, ranked in the k-th position of R. Being the first means that all the items R_1, R_2,…, R_{k-1} above R_k are not recommendable because they do have a training rating. Let $train_j$ represent the event that R_j has a training rating by the target user, that is $train(u, R_j) = 1$. Similarly, let rel_j mean R_j is relevant, and so forth for the random variables $test$ and $train$. If we marginalize $p(P@1 = 1|R)$ by the possibility that the k-th item is the first recommendable, we have:

$$\mathbb{E}[P@1|R] = \sum_{k=1}^{n} p(rel_k, train_1, \dots, train_{k-1}, \neg train_k|R)$$

where $n = |\mathcal{I}|$ is the total number of items in the system.

We can follow an analogous development for observed precision, for which we get:

$$\mathbb{E}\left[\hat{P}@1\middle|R\right] = \sum_{k=1}^{n} p(rel_k, test_k, train_1, \dots, train_{k-1}|R)$$

where we can remove the condition $\neg train_k$ because it follows from $test_k$ – if a rating is present in the test set it cannot be present in the training set.

We shall now assume that rating one item is independent from rating others, i.e. $p(rated_1, \dots, rated_{k-1}|R) = \prod_{j=1}^{k-1} p(rated_j)$. This is not necessarily true, since users

might tend to rate items with some specific characteristics. However, we reasonably assume that the possible imprecisions of this simplification may affect all algorithms similarly, and therefore, it does not affect the validity of our conclusions.

This item rating independence assumption, combined with the random split protocol – where ratings are independently sampled –, leads $train$ and $test$ variables to inherit the independence. Thus, recovering the notation $p(rel|R_k)$ for $p(rel_k)$ and same for $train$ and $test$, we have:

$$\mathbb{E}[P@1|R] \sim \sum_{k=1}^{n} p(rel, \neg train|R_k) \prod_{j=1}^{k-1} p(train|R_j) \tag{4.2}$$

$$\mathbb{E}[\hat{P}@1|R] \sim \sum_{k=1}^{n} p(rel, test|R_k) \prod_{j=1}^{k-1} p(train|R_j) \tag{4.3}$$

The above expressions represent the expected precision of a generic ranking and are the starting point for the rest of our analysis. Their most novel aspect is that they explicitly consider exclusion of training ratings, a key characteristic that distinguishes recommender systems from other areas of information retrieval. We point out such aspect of the formulas in more detail in the following section, and compare them with similar results of IR. Note that we intentionally remove the variable that represents the user u for simplicity, but it is implicit in the condition of all the probabilities since, for now, the ranking R is a ranking offered to a specific target user.

In the following sections we use the previous expressions to derive the ranking criterion that maximizes each expected precision and study the effectiveness obtained by popularity-based recommendations under different situations.

4.3 Optimal recommendation

Using equations 4.2 and 4.3 we can deduce the optimal criterion we must use to rank the items in order to obtain the maximum possible precision. We enunciate such result by means of the following lemma.

Lemma 1. Assuming item rating independence, the optimal recommendation R that maximizes the expected true precision ($\mathbb{E}[P@1|R]$) under a random rating split, ranks items by non-increasing value of:

$$f(k) = p(rel|\neg train, R_k) = p(rel|R_k) \frac{1 - \rho\, p(rated|rel, R_k)}{1 - \rho\, p(rated|R_k)} \tag{4.4}$$

Under the same assumptions, the optimal recommendation R that maximizes the expected observed precision ($\mathbb{E}[\hat{P}@1|R]$) ranks items by non-increasing value of:

$$\hat{f}(k) = \frac{p(rel, test|R_k)}{p(\neg train|R_k)} \propto p(rel|R_k)\frac{p(rated|rel, R_k)}{1 - \rho\, p(rated|R_k)} \tag{4.5}$$

Proof. Given any function g over items, any ranking can be generated from the ordered one – i.e. ordered in decreasing order of g – by a sequence of swaps of adjacent items, let say R_k and R_{k+1}, where $g(R_k) > g(R_{k+1})$ (see for instance the proof of correction of bubble sort). In order to show that the above rankings maximize the corresponding precision, it is therefore enough to show that a swap against f or $\hat{f}$ in a ranking produces a smaller value for $\mathbb{E}[P@1|R]$ or $\mathbb{E}\left[\hat{P}@1\middle|R\right]$ respectively.

For true precision, let R be some ranking so that $f(k) \geq f(k+1)$ for some k, and let us consider a ranking R' resulting from swapping R_k and R_{k+1} in R. Using equation 4.2 we have that the expected true precision of both rankings is:

$$\begin{aligned}
&\mathbb{E}[P@1|R] \\
&\quad = C_1 + p(rel, \neg train|R_k)\; C_2 + p(rel, \neg train|R_{k+1})p(train|R_k)\; C_2 + C_3 \\
&\mathbb{E}[P@1|R'] \\
&\quad = C_1 + p(rel, \neg train|R_{k+1})\; C_2 + p(rel, \neg train|R_k)p(train|R_{k+1})\; C_2 + C_3
\end{aligned}$$

where C_1, C_2 and C_3 are terms that do not depend on R_k or R_{k+1}.

The difference between both rankings is therefore:

$$\begin{aligned}
\mathbb{E}[P@1|R] - \mathbb{E}[P@1|R'] & \\
&\propto p(rel, \neg train|R_k)\big(1 - p(train|R_{k+1})\big) \\
&\quad - p(rel, \neg train|R_{k+1})\big(1 - p(train|R_k)\big)
\end{aligned}$$

and then it is easy to see that:

$$\begin{aligned}
\mathbb{E}[P@1|R] \geq \mathbb{E}[P@1|R'] &\Leftrightarrow \frac{p(rel, \neg train|R_k)}{1 - p(train|R_k)} \geq \frac{p(rel, \neg train|R_{k+1})}{1 - p(train|R_{k+1})} \\
&\Leftrightarrow f(k) \geq f(k+1)
\end{aligned}$$

which is true by description of R. That is, swapping R_k and R_{k+1} decreases $\mathbb{E}[P@1|R]$. And an analogous reasoning proves the corresponding statement for observed precision, by substituting $\neg train$ by $test$.

The right-side form of f and $\hat{f}$ in equations 4.4 and 4.5 is obtained by applying $p(train|i) = \rho\; p(rated|i)$ and $p(test|i) = (1 - \rho)\; p(rated|i)$. □

Lemma 1 states that the optimal non-personalized recommendation for true precision is obtained by decreasing probability of relevance among non-training ratings $p(rel|\neg train, i)$. This probability corresponds to the fraction of unobserved (unrated) user tastes that are positive: the ratio of positive missing ratings. This means that the best items to be recommended are the ones for which most unobserved preferences are positive. We shall refer to this statement as Low prior Discovery Recall Principle (LDRP),

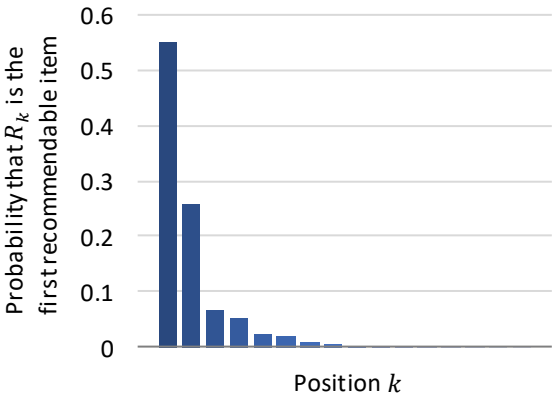

Figure 4.1. Fraction of users for who the first recommendable item is ranked in the k-th position of the ranking or, in other words, the probability that R_k is the first recommendable item. The ranking is ordered by relevant popularity using the data of MovieLens.

since the lower the fraction of relevant preferences is discovered (and thus rated) the better for an item in order to be recommended (Cañamares and Castells 2018b).

Such statement implies a revision of the Probability Ranking Principle (PRP) from information retrieval to the context of recommender systems. PRP is an important principle proposed and discussed by Robertson in 1977. It is framed in the area of information retrieval systems, systems whose main task consists on ranking a collection of documents according to an information need (e.g. a query). It states that under certain assumptions, the optimal ranking consists on sorting de documents by decreasing probability of relevance to the information need.

If we literally translate PRP to recommender systems, it implies that items must be ranked by decreasing probability of relevance $p(rel|i)$ in order to maximize true accuracy, instead of by $p(rel|\neg train, i)$ as our LDRP states. The key of this disagreement is the exclusion of rated items from the final recommended rankings, a particularity of recommender systems that is not considered in PRP, since already visited documents are not excluded in information retrieval systems.

Such particularity can indeed produce significant modifications of the final ranking offered to each user, as we illustrate in Figure 4.1. It shows how far item exclusion can go in the relevant popularity ranking on MovieLens, by indicating the fraction of users for who the first recommendable item is in the k-th position of such ranking. We see that the first ranked item is recommended only for approximately 55% of the users, namely, it is excluded for the other 45% because they have already rated it. And more than 15% of users have indeed rated the first two items (the first two bars of the graph sum less than 0,85). To some users, exclusion even reaches the seventh or eighth position of the ranking. Therefore, this particular aspect of recommender systems, that we formally include in our

analysis, might make a difference in the computation of the metrics and even the outcome of a comparative evaluation of algorithms.

Another aspect of the optimal rankings to be mentioned is that they can be applied to both personalized and non-personalized levels. That is, we may consider the expected precision of offering the ranking R to a specific user u: $\mathbb{E}[P@1|R,u]$, and re-express Lemma 1 in terms of $p(rel|R_k,u)$, $p(rated|R_k,u)$ and $p(rated|rel,R_k,u)$. The optimality statements of Lemma 1 will thus refer to the best potential ranking that can be offered to each user u.

That is however not the case with popularity-based ranking functions, since they are non-personalized recommendations and therefore do not consider a specific ranking for each user. As we have said before, relevant popularity ranks items by $p(rated,rel|R_k)$, but not specifically by $p(rated,rel|R_k,u)$ for each user. Therefore, in order to compare both optimal rankings and different popularity variants in several situations, we shall reduce the scope of Lemma 1 to non-personalized methods (i.e. we only consider recommendation algorithms where R does not depend on u). In other words, in the following sections when we say "optimal recommendation algorithms" we mean "between all those algorithms that offer the same ranking R to all users".

In order to help the reader to visualize and mentally organize all the situations we will study in the following sections, Tables 4.1 and 4.2 – in page 49, just before Section 4.8 – provide a summary with the independence conditions that describe each of them. Table 4.3 includes the relative observed and true effectiveness (optimal, greater than random, equal to random or the worst possible) that each popularity-based recommendation obtains in each scenario. The column Label identifies all the situations and is provided to combine this Table 4.3 with the previous Tables 4.1 and 4.2. Complementarily, Table 4.4 shows the ranking functions that both optimal rankings and popularity variants present in each situation.

4.4 Relevance-independent rating bias

Before considering more complex situations of dependency between users' tastes and how they rate, we start by analysing what happen in a neutral situation when this dependency does not exist – formally $p(rated|rel,i) \sim p(rated|i)$. This is generally not the case in common settings but allows to set out some optimality properties of popularity and average rating, as well as study in isolation the effect of the popularity rating bias.

With this independence assumption between $rated$ and rel (given the item), the different popularity ranking criteria formulated in Section 4.1.2 (Equation 4.1) result in:

$$pop(R_k) \propto p(rated|R_k)$$
$$rpop(R_k) \propto p(rated|R_k)\, p(rel|R_k)$$
$$avg(R_k) \sim p(rel|R_k)$$

Applying Lemma 1 we have that in true precision the maximum value is obtained raking the items by decreasing value of relevance probability:

$$f(k) \sim p(rel|R_k)\frac{1-\rho\, p(rated|R_k)}{1-\rho\, p(rated|R_k)} = p(rel|R_k)$$

While for observed precision the expression results in:

$$\hat{f}(k) \sim p(rel|R_k)\frac{p(rated|R_k)}{1-\rho\, p(rated|R_k)} \propto p(rel|R_k)\, p(rated|R_k)$$

where the rank equivalence holds because $g(x) = x/(1-\rho\, x)$ is a monotonically increasing function in $x \in (0,1)$ and almost equal to the identity function for small values of x. Along this thesis we will used the symbol $\propto$ to denote "monotonically increasing with" and not necessarily "proportional to", since the monotonicity is enough to ensure that two criteria produce the same resulting ranking when using them to sort the items.

Returning to the previous formulas, note first that observed and true precisions do not necessarily agree when the rating probability does not depend on relevance. Regarding to the performance of the different popularity-based recommendations, comparing the previous five ranking functions we may conclude the following lemmas.

Lemma 2. Assuming independence between rating and relevance of items, and a random rating split, we have that:

- Average rating is the optimal non-personalized recommendation in $P@1$ (true precision).
- Relevant popularity is optimal in $\hat{P}@1$ (observed precision).
- Total popularity is equivalent to random recommendation in $P@1$ (true precision).

Proof. The first two statements about average rating and relevant popularity may be directed inferred from the comparison between their ranking functions and the optimal ones since $avg(R_k) \sim f(k)$ and $rpop(R_k) \propto \hat{f}(k)$. The equivalence between total popularity and random recommendation is due to that, under the independence assumption between rating and relevance, $p(rated|R_k)$ is unrelated to $p(rel|R_k)$, and so are $pop(R_k)$ and $f(k)$. □

Note again that the previous optimality conclusions are framed in a non-personalized context. So, of course, personalized methods can obtain greater accuracy, but any other non-personalized algorithm will be equal or worse.

With additional conditions, the following lemma states properties for the true precision of relevant popularity, and the observed precision of total popularity.

Lemma 3. Assuming independence between rating and relevance of items, and a random rating split, we consider two situations.

If the relevance distribution over items is steeper enough than the rating distribution (i.e. relevance distribution dominates the product of both distributions), then:

- Relevant popularity is optimal in terms of $P@1$ (true precision).
- Average rating is optimal in $\hat{P}@1$ (observed precision).
- Total popularity is equivalent to random recommendation in $\hat{P}@1$ (observed precision).

If on the contrary, ratings are steeper enough than relevance, then:

- Relevant popularity is equivalent to random in $P@1$ (true precision).
- Average rating is equivalent to random in $\hat{P}@1$ (observed precision).
- Total popularity is optimal in $\hat{P}@1$ (observed precision).

Proof. First, if the relevance distribution is steeper enough than the rating distribution, $p(rel|R_k)$ would dominate over $p(rated|R_k)$ when multiplying them, and we would have:

$$\hat{f}(k) \propto p(rel|R_k)p(rated|R_k) \propto p(rel|R_k)$$
$$rpop(R_k) \sim p(rel|R_k)p(rated|R_k) \propto p(rel|R_k)$$

Then it holds that $rpop(R_k) \propto f(k)$ and $avg(R_k) \sim \hat{f}(k)$, whereas total popularity $pop(R_k) \propto p(rated|R_k)$ would be unrelated to $\hat{f}(k)$ and therefore equivalent to random recommendation in observed precision.

If on the contrary the rating distribution is steeper enough than relevance, we would have:

$$\hat{f}(k) \propto p(rel|R_k)p(rated|R_k) \propto p(rated|R_k)$$
$$rpop(R_k) \sim p(rel|R_k)p(rated|R_k) \propto p(rated|R_k)$$

Total popularity would be optimal in observed precision since $pop(R_k) \propto \hat{f}(k)$; average rating $avg(R_k) \sim p(rel|R_k)$ would be unrelated to $\hat{f}(k) \propto p(rated|R_k)$ and therefore equivalent to a random ranking in observed precision; and the relevant popularity ranking by $rpop(R_k)p(rated|R_k)$ would be unrelated to $f(k) \sim p(rel|R_k)$ and therefore random in true precision. □

As a corollary, we can also see than in the intermediate case where neither or both distributions dominates the product $p(rel|R_k)p(rated|R_k)$, relevant popularity should be sub-optimal but still better than random in true precision, and average rating and total popularity can be expected to be greater than random in observed precision.

All the previous situations are described in Table 4.1 and the corresponding conclusions can be found in Table 4.3 (label a). We appreciate here, for the first time, that average rating seems to provide further guarantees in true precision than relevant popularity, and certainly than total popularity. However, in observed precision, relevant popularity is optimal whereas average rating is not. Actually, looking at label a of Table 4.3, we shall notice

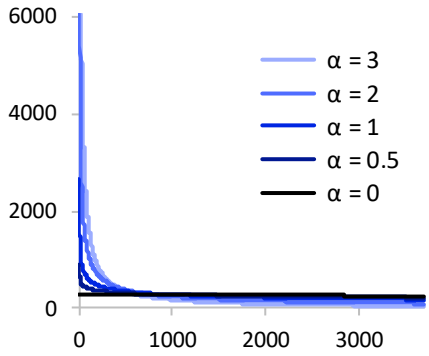

Figure 4.2. Item rating distributions (i.e. number of ratings that each item has received) resulting from the simulation of the rating process described in Section 4.4.1, for different values of the bias popularity parameter (α).

a symmetry: the behaviour of average rating in true precision is exactly the same that the behaviour of relevant popularity in observed precision, an the same applies to the observed precision of average rating and the true precision of relevant popularity. This means that even though (relevant or total) popularity might display a noticeably better accuracy than average rating in a standard offline experiment, recommending by average rating might be actually better. However, so far this is true under a specific simplifying assumption.

4.4.1 Influence of the popularity distribution bias

We now verify, both theoretically and empirically, the direct impact that popularity biases of the rating data have on the observed effectiveness of popularity. Doing it at this point, under the relevance independence assumption, allows us to study their effects in isolation, removing the potential impacts of rating relevant and non-relevant items to different extent.

In the proof of Lemma 1 referring observed precision, we prove that $\mathbb{E}[\hat{P}@1|R]$ decreases when we swap two consecutive items of R – let say they are in the positions k and $k+1$, respectively – if they meet the condition $\hat{f}(k) \geq \hat{f}(k+1)$. And we therefore conclude that $\hat{f}(k)$ is the criterion we must rank with in order to obtain the maximum expected observed precision.

As a corollary, we can realize that the larger the difference $\hat{f}(k) - \hat{f}(k+1)$, the bigger the loss in $\mathbb{E}[\hat{P}@1|R]$ when swapping both items, and hence the greater the accuracy advantage of the optimal ranking over any other non-personalized alternative. But, under the relevance and rating independence assumption, we show above that $\hat{f}(k)$ coincides precisely with $rpop(R_k)$, since both are equivalent to rank by $p(rel|R_k)\,p(rated|R_k)$. Relevant popularity is therefore not only the optimal ranking in observed precision, but also its distribution bias determines how large is the difference

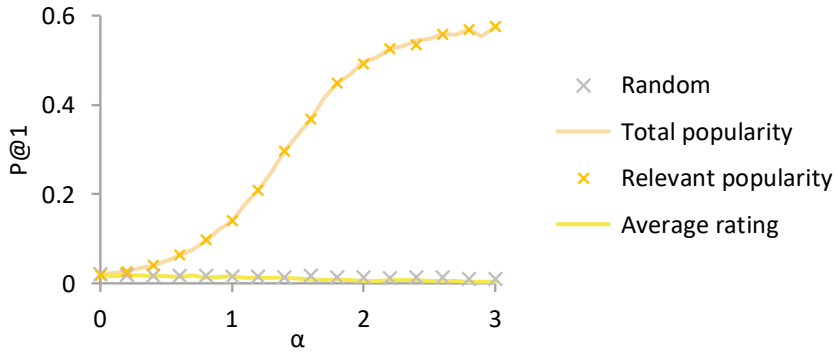

Figure 4.3. Evolution of the observed precision for different recommenders – random, total and relevant popularity and average rating – as a function of the item distribution bias (α).

$\hat{f}(k) - \hat{f}(k+1)$ and, consequently, its own advantage over other recommendation algorithms. In short, relevant popularity is underpinned by rating distribution biases and would, on the other hand, become effectless (i.e. equivalent to random recommendation) in their absence.

To illustrate the previous result, we conduct a simulation where we generate several rating distributions varying the item popularity bias. To this end, we use the dimensions – number of users, items and relevant ratings – of the MovieLens dataset described in Section 2.3. Such dimensions can be found in Table 2.1.

We consider two probability distributions in order to generate the ratings, one for users and other for items, which indicate the probability that certain user/item is the one who does/receives the next rating. Thus, the probability that a specific user rates a particular item is computed as the product between these two previous distributions. According to this last probability we select user-item pairs without replacement. When a pair is selected, we assig a rating to such pair. The rating preference is randomly decided with a certain provability – $p(rel)$ – that we estimate from the number of relevant ratings in MovieLens. Thereby relevance is independent from the rest of variables, particularly from rating. Note that with this setting relevance probability is constant on items – $p(rel|R_k) \sim p(rel)$ – and therefore both popularities rank by $p(rated|R_k)$, whereas average rating is equivalent to the random recommendation.

We model both item and user probability distributions with a Pareto distribution. According to such distribution, the probability of the k-th user or item is determined by the formula $g(k) = C\ k^{\alpha}$ where the constant C takes the appropriate value to sum 1 over items (or users). Regarding the exponent α, a high value means a larger distribution bias. In the simulation, the exponent of the user probability distribution takes always the value 3. Regarding the item distribution, we vary such exponent between 0 and 3 in order to generate different-biased distributions and be thus able to analyse the influence of such

bias. Figure 4.2 shows distinct item rating distributions resulting from the simulation process described above, for different values of the bias parameter (α).

Using these distributions, we run and evaluate the recommendation algorithms – random, total and relevant popularity and average rating – through a random split of $\rho = 0.8$. Figure 4.3 displays the observed precision evolution of the previous algorithms as a function of the item distribution bias – defined by the Pareto distribution exponent.

We first conclude that the observed precision of both total and relevant popularity coincides, and so do average rating and random, as we expected. The precision of the latter is essentially constant and does not depend on the bias. However, popularity precision increases with the bias, and so does the difference with random, as we predicted in the analytical development.

We have therefore seen how observed precision wrongly rewards (relevant and total) popularity in a setting when all methods must be at random-level in terms of true precision, since $f(k) \sim p(rel|R_k)$ is constant on items and completely independent from the other variables. In contrast, average rating seems better to reflect reality, at least in this scenario.

4.5 The interplay between rating and relevance

Assuming independence between rating and relevance has allowed us to verify the isolated effect of popularity bias, as well as conclude that average rating results in a more reliable option than relevant popularity – in terms of true precision – when considering a neutral scenario. Such scenario does not however represent a common real environment, since it is well-known that user's tastes have a remarkable influence in the rating generation process (Marlin et al. 2007).

Here we thus consider a more generic and representative situation, where a dependence between relevance and rating exists. In order to characterize and analyse the effects of such dependence in the effectiveness of the popularity-based recommendations, we shall study the process that give rise to a rating and identify its steps. Depending whether relevance comes into play at some stages or others, we will have different situations to analyse.

For a user to rate an item, she must first of all discover it somehow. The item discovery distribution may become determinant in the rating distribution, since the more discovered the item, the more likely to be rated, and therefore to reach the top positions of the popularity ranking. This discovery can be carried out through a wide range of means (search engines, advertising, social networks, recommender systems, etc.) and is essentially fortuitous since, though the user can chose the means to get information from the world, she cannot obviously determine in advance which items are going to be discovered. Nevertheless, certain tendency to find relevant items can be considered, since we used to meet

people with similar tastes and search information in places that usually show things of our interest. Moreover, we benefit from the capacity of tools as search engines, browsing interfaces or recommender systems themselves, to find information that satisfies our interest.

After discovering the item, the user needs to consume it in order to form an opinion about it. For instance, if it is a film, she must watch it, if it is a book, read it, if it is a song, listen to it, etc. For simplicity, we will collapse discovery and consumption as a single event, as if users instantaneously form an opinion about items when discovering them.

Finally, after the discovery step, the user needs to decide whether to enter a rating reflecting her level of appreciation for the item or not. Decision that is usually influenced by such appreciation level. As we pointed out in Chapter 3 when introducing the related work of this thesis, common public datasets show that user's rating behaviour is often biased towards reflecting positive preferences.

The rating distribution over items is therefore the result of a discovery distribution, followed by a rating decision distribution over discovered items. The general dependence between relevance and rating derives thus from the influence of the user's tastes in each of these two distributions, and can be therefore studied separately. Depending on whether such influence takes place at discovery or rating decision level (or both) it will give rise to some situations or others.

In order to carry out the study of such situations, we reflect in the formulas this decomposition of the rating process by introducing the binary random variable $seen: \mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$ that takes value 1 for a user-item pair if the user knows the item exists, and 0 otherwise. Given that rating an item necessarily implies having discovered it – i.e. $p(rated|\neg seen, i) = 0$ – we have:

$$p(rated|i) = p(rated, seen|i) = p(rated|seen, i)p(seen|i)$$

Introducing this into equations 4.4 and 4.5, and marginalizing by relevance, we get:

$$f(k) = \frac{p(rel|R_k)(1 - \rho\ a_k)}{1 - \rho\ b_k + \rho\ (b_k - a_k)\ p(rel|R_k)} \tag{4.6}$$

$$\hat{f}(k) = \frac{a_k\ p(rel|R_k)}{1 - \rho\ b_k + \rho\ (b_k - a_k)\ p(rel|R_k)} \tag{4.7}$$

where the terms a_k and b_k have the following expressions:

$$a_k = p(rated|seen, rel, R_k)p(seen|rel, R_k)$$
$$b_k = p(rated|seen, \neg rel, R_k)p(seen|\neg rel, R_k)$$

Repeating the process for the ranking functions of the different popularity-based recommendations (Equation 4.1) we obtain that total popularity ranks items according to:

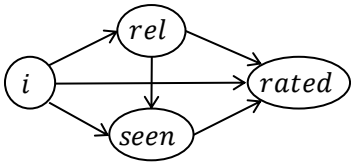

Figure 4.4. Probabilistic graphical model (bayesian network) for the random variables involved in the generation of ratings.

$$\begin{aligned} pop(R_k) &\propto p(rated|R_k) \\ &= p(rated|seen, rel, R_k)p(seen|rel, R_k)p(rel|R_k) \\ &\quad + p(rated|seen, \neg rel, R_k)p(seen|\neg rel, R_k)\big(1 - p(rel|R_k)\big) \\ &= (a_k - b_k)\, p(rel|R_k) + b_k \end{aligned} \tag{4.8}$$

Relevant popularity by:

$$\begin{aligned} rpop(R_k) &\propto p(rated, rel|R_k) \\ &= p(rated|seen, rel, R_k)p(seen|rel, R_k)p(rel|R_k) \\ &= a_k\, p(rel|R_k) \end{aligned} \tag{4.9}$$

And finally, the ranking function of average rating is the quotient between the previous ones.

$$avg(R_k) \sim \frac{p(rated, rel|R_k)}{p(rated|R_k)} \propto \frac{a_k\, p(rel|R_k)}{(a_k - b_k)\, p(rel|R_k) + b_k} \tag{4.10}$$

We can thus realize that the ranking function of both the optimal rankings and the popularity-based recommendations are fully described by three distributions: the relevance distribution over items $p(rel|R_k)$, the discovery distribution – represented by the terms $p(seen|rel, R_k)$ and $p(seen|\neg rel, R_k)$ – and the rating decision distribution – characterized by the terms $p(rated|seen, rel, R_k)$ and $p(rated|seen, \neg rel, R_k)$. Note that both discovery and rating decision distributions reflect a potential bias towards discover/rate some items more than others – depending on the relevance of the item and/or the item itself. These are what we refer and study in the next sections as discovery bias and user behaviour bias, respectively.

Figure 4.4 shows the Bayesian network reflecting the situation described above, with all the potential probabilistic dependencies between the random variables.

Before analysing the effect of the user behaviour and discovery biases, we set out a mathematical statement useful to derive some of the conclusions of the following sections.

Statement 1. The function $g(x) = x/(c_1 + c_2\, x)$ is a monotonically increasing function of x as long as $c_1 > 0$, whatever the value of c_2.

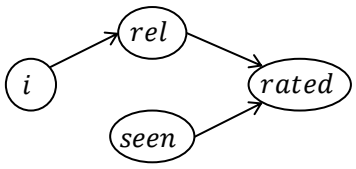

Figure 4.5. Bayesian network reflecting the neutral discovery assumption of the user behavior study.

4.6 User behaviour bias

Because of the spontaneous user's rating decision, ratings collected from real scenarios usually present a missing not at random distribution (MNAR), as we explain before when describing related work in Chapter 3. One of the biases of such distribution is related with the sign of the manifested preferences, namely, positive ratings are usually more frequent than negative ones. We provide here a formal characterization and analysis on the consequence of such trend, an analysis that also covers the opposite situation (where users rate more often the non-relevant items).

We start by making a general simplifying assumption: relevance is the main intrinsic property of an item that governs the user's rating decision. In other words, the decision is conditionally independent from the item given its relevance: $p(rated|seen, rel, i) \sim p(rated|seen, rel)$ and $p(rated|seen, \neg rel, i) \sim p(rated|seen, \neg rel)$. This is not a far-fetched simplification. Remind from the explanation of the related work that, in a real survey conducting by Marlin et al (2007), a great majority of users expressed that their opinion on items significantly affects their decision on rating them.

Both previous parameters $p(rated|seen, rel)$ and $p(rated|seen, \neg rel)$ represent the users' tendency to rate what they like and dislike, respectively. They thus characterize the user behaviour bias and allows us to describe different situations. In order to analyse such situations, we are therefore interested in studying how these parameters influence in the expected precision.

For that purpose, we consider a neutral behaviour in the variables we are not analysing, that is, we consider a neutral discovery that does not depend on relevance nor items: formally $p(seen|rel, i) \sim p(seen)$. The Bayesian network reflecting these and the previous independence assumptions (i.e. neutral discovery and item independent rating decision) is depicted in Figure 4.5.

Under such assumptions the terms a_k and b_k of Equations 4.6 and 4.7 are constants that do not depend on R_k, so we redefine them as a and b:

$$a_k = p(rated|seen, rel)\, p(seen) = a$$
$$b_k = p(rated|seen, \neg rel)\, p(seen) = b$$

And then, both optimal rankings (Equations 4.6 and 4.7) reduce to sorting by decreasing probability of relevance:

$$f(k) \propto \hat{f}(k) \propto \frac{p(rel|R_k)}{c + d\, p(rel|R_k)} \propto p(rel|R_k) \tag{4.11}$$

where $c = 1 - \rho\, b$ and $d = \rho\, (b - a)$. The last step is a consequence of the Statement 1, which we can apply here since c is always positive.

Note that, in this situation when we are considering no discovery bias, observed precision is consistent with true precision.

Applying now the independence assumptions to the popularity ranking functions (Equations 4.8, 4.9 and 4.10) we have:

$$\begin{aligned} pop(R_k) &\propto (a - b)\, p(rel|R_k) + b \\ rpop(R_k) &\propto a\, p(rel|R_k) \\ avg(R_k) &\propto \frac{a\, p(rel|R_k)}{(a - b)\, p(rel|R_k) + b} \end{aligned} \tag{4.12}$$

That means that relevant popularity becomes also proportional to the probability of relevance, and then is optimal in both observed and true precision.

For total popularity, the order depends on the sign of the difference $a - b \propto p(rated|seen, rel) - p(rated|seen, \neg rel)$. We distinguish three potential situations. If there is no difference, that is, $p(rated|seen, rel) = p(rated|seen, \neg rel)$, users rate relevant items as often as irrelevant ones and then user behaviour does not depend on relevance. In such case, all items present the same number of ratings ($pop(i) \propto b$) and total popularity is equivalent to random recommendation. If users are more inclined to rate items they like than items they do not – i.e. $p(rated|seen, rel) > p(rated|seen, \neg rel)$ – total popularity ranks items by relevance and is therefore optimal in both observed and true precision. If, on the contrary, users are biased towards rating items that they do not like – $p(rated|seen, rel) < p(rated|seen, \neg rel)$ – then total popularity is the least accurate recommendation possible (it sorts the items in reverse order to the one we must follow in order to obtain the maximum precision).

Regarding average rating, we can apply Statement 1 to Equation 4.12 and claim that it ranks by relevance and is therefore optimal. This is true as long as $b \propto p(rated|seen, \neg rel) > 0$. Thus, provided that non-relevant items are rated to some extent, average rating should behave exactly as well as relevant popularity. That is the case in most common situations, but we should remark that if $p(rated|seen, \neg rel) = 0$ then the ranking function becomes a constant. This makes sense, since this condition means that there are no negative ratings in the dataset, and then average rating (in its binary definition) is the same for all items and does not make sense as a recommendation criterion. This would be the case, for instance, in recommendation on implicit data, where all feedback is positive.

To sum up, and excluding atypical situations of only positive ratings, we observe that relevant popularity and average rating are both robust to unusual user behaviour – they are optimal even when non-relevant ratings are more frequent. Total popularity, by contrast, may find its effectiveness affected in such situation, since the most rated items are precisely the less relevant ones.

4.7 Discovery bias

Now, in order to study in isolation the specific effects of discovery biases, we assume a neutral user behaviour, much like we have done with discovery in the previous section. Formally we are considering that rating decision is independent from relevance and items: $p(rated|seen, rel, i) \sim p(rated|seen, \neg rel, i) \sim p(rated|seen)$.

When studying the user behaviour bias in previous section, we consider relevance as the main item property that leads the user to rate it, and we thus removed the explicit item dependency of the rating decision. In this situation, however, it does not seem sensible to eliminate the dependency between discovery and item, since discovery is the result of several complex processes, some of which do not actually treat all the items in the same way. Advertising campaigns, for instance, may have special interest in promoting some specific products, giving rise to an unequal level of information diffusion that facilities the discovery of some items above others. Relevance dependency cannot be removed either, since it is the main factor that affects discovery when it takes place through search engines, recommender systems or the suggestion of a friend, for instance.

In order to study the effect of these two potential sources of biases (relevance and items), we first consider each of them in isolation, assuming a neutral distribution in the other. After that, we will discuss what can we expect in situations where discovery depends on both relevance and items at the same item.

4.7.1 Relevance discovery bias

Dependency between discovery and relevance represents the ability of the user – together with search engines, recommender systems, etc. – to discover what she likes. We start by analysing the case where such ability is the only factor which determines what it is discovered. Any other characteristic of the item is therefore not considered, as how known it already is or the interest of some company in advertising it. In this situation, all relevant items tend to be discovered to the same extend, and the same shall applies to non-relevant ones. Formally, we are considering that discovery is conditionally independent from the specific item given its relevance: $p(seen|rel, i) \sim p(seen|rel)$ and $p(seen|\neg rel, i) \sim p(seen|\neg rel)$. This situation corresponds with the Bayesian network displayed in Figure 4.6.a.

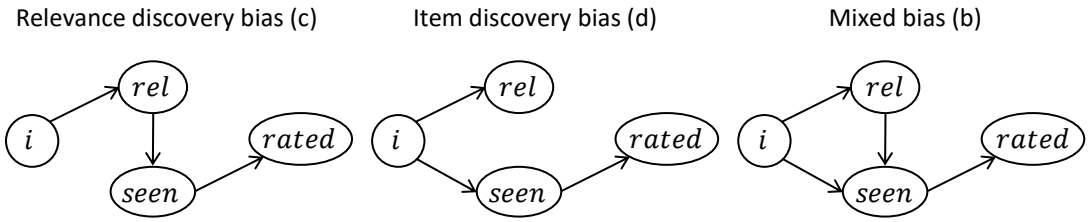

Figure 4.6. Bayesian networks of the different situations that arise in the discovery bias study. The labels c and d in the graph match the labels of Tables 4.1 to 4.4. Case b does not appear in such tables, but it will be referred with this label in the experiments of Section 4.8.2.

Under this approach, the different situations are characterized by the values of $p(seen|rel)$ and $p(seen|\neg rel)$, i.e. by whether relevant items are discovered more or less often than non-relevant ones. The study of such situations follows a quite similar structure than the one followed in the user behaviour analysis of Section 4.6. The reason is that, under these assumptions, the terms a_k and b_k of Equations 4.6 and 4.7 are once again constant with respect to R_k.

$$a_k = p(rated|seen)\, p(seen|rel) = a$$
$$b_k = p(rated|seen)\, p(seen|\neg rel) = b$$

And then ranking functions of both optimal rankings and popularity recommendations keep the same form than in Equations 4.11 and 4.12, but using the previous values for a and b. Consequently, observed precision agrees once again with true precision, and both optimal rankings sort items according to the probability of relevance. So is the case of relevant popularity, which is therefore optimal in both precisions.

Total popularity depends on the sign of the difference $a - b \propto p(seen|rel) - p(seen|\neg rel)$, namely, of whether users discover more what they like that what they dislike or not. We can thus distinguish three situations depending of the value of the parameters $p(seen|rel)$ and $p(seen|\neg rel)$. If $p(seen|rel) = p(seen|\neg rel)$, discovery does not depend on relevance and, since neither does on items, it is independent from all the variables. In such case, all items are discovered – and thus rated – to the same extent and then the number of ratings is not an informative signal, making total popularity equivalent to random recommendation. If $p(seen|rel) > p(seen|\neg rel)$, relevant items are more discovered that non-relevant ones. In this situation, total popularity ranks items by relevance and is therefore optimal in both observed and true precision. Finally, if $p(seen|rel) < p(seen|\neg rel)$ discovery is biased towards non-relevant items, and then total popularity ranks contrary to relevance probability producing the worst possible ranking.

Regarding average rating, the condition $b > 0$ must be met in order to apply Statement 1 and conclude that the quotient of Equation 4.12 is equivalent in ranking to relevance probability. Since $b = p(rated|seen)\, p(seen|\neg rel)$, we need

$p(rated|seen) > 0$ – this is obviously true or we would not have ratings – and $p(seen|\neg rel)$, i.e., we need that non-relevant items are discovered in some proportion distinct from 0, a condition that we may reasonably give for granted.

4.7.2 Item discovery bias

We now consider the case where the probability of discovery depends only on each specific item, regardless of its relevance. Dependency between discovery and item reflects the effort of different items – or more precisely of those who create, commercialize, or advertise them – to make then known by the largest number of users. Removing the dependency with relevance means that this effort is highly skewed between items, to the point where a difference of relevance (quality, utility, etc.) barely plays a perceptible role.

Formally this means we assume the conditional independence $p(seen|rel, i) \sim p(seen|\neg rel, i) \sim p(seen|i)$ as depicted in Figure 4.6.b. Under this assumption, plus neutral user behaviour, we have that both terms a_k and b_k (appearing in Equations 4.6 to 4.10) coincide. However, they are not constant in this situation:

$$a_k = b_k = p(rated|seen)p(seen|\ R_k)$$

Substituting the value of such terms in Equations 4.6 and 4.7 we obtain that optimal rankings sort the items according to the following functions:

$$f(k) \sim p(rel|R_k)$$

$$\hat{f}(k) = p(rel|R_k)\,\frac{a_k}{1-\rho\ a_k} \propto p(rel|R_k)\ p(seen|R_k)$$

where last step of the observed optimal ranking $\hat{f}(k)$ holds because $a_k/(1-\rho\ a_k) \propto a_k$ by Statement 1, and $a_k \propto p(seen|R_k)$. Repeating the process for the popularity rankings (Equations 4.8, 4.9, and 4.10) we have:

$$pop(R_k) \propto p(seen|\ R_k)$$

$$rpop(R_k) \propto p(seen|R_k)\ p(rel|R_k)$$

$$avg(R_k) \sim p(rel|R_k)$$

Note that in this situation, under the assumption of a relevance-independent discovery bias, we also have:

$$\begin{aligned} p(rated|rel, i) &= p(rated|seen, rel, i)\ p(seen|rel, i) \\ &\sim p(rated|seen, i)\ p(seen|i) = p(rated|i) \end{aligned}$$

Therefore, we are in the situation studied in Section 4.4 where rating and relevance were independent. We can thus apply Lemma 2 and conclude that average rating is optimal in true precision, relevant popularity is optimal in observed precision, and total popularity is random in true precision.

In order to apply Lemma 3 we shall note that $p(rated|i) \sim p(rated|seen)$ $p(seen|i) \propto p(seen|i)$, and therefore the statements of such lemma referring rating distribution can be applied to the discovery one. Consequently, and in accordance to Lemma 3, the true precision of relevant popularity and the observed precision of average rating and total popularity will depend on who dominates the product of relevance and discovery distributions: $p(seen|R_k)$ or $p(rel|R_k)$.

Thus, relevant popularity will tend to be optimal in true precision if $p(rel|R_k)$ is steeper enough than $p(seen|R_k)$, and equivalent to random if it is $p(seen|R_k)$ who dominates the product. In an average case where it is not clear which distribution is steeper, relevant popularity would be not optimal but still better than random. Average rating follows the same structure but for observed precision. Namely, it will be optimal in observed precision if $p(rel|R_k)$ dominates the product and tend to random if $p(seen|R_k)$ does. In a situation in between, average rating would be better than random although not optimal. Finally, total popularity will tend to random in observed precision if $p(rel|R_k)$ is steeper enough than $p(seen|R_k)$, and will be optimal in the opposite situation. In the intermediate case, it will be greater than random but not optimal.

Tables 4.1, 4.2, 4.3 and 4.4 summarize each of the situations studied so far by describing their corresponding independence conditions and the effectiveness obtained by each of the popularity-based recommendations on both observed and true precision. Taking a general overview of Table 4.3, we observe that average rating is always a more reliable option than relevant popularity in terms of true precision, while observed precision is telling exactly the opposite message. Moreover, consulting Table 4.4 we can note that sorting by relevance probability always generate the best potential ranking in terms of true precision. All this is true however to the extent that the independence conditions of each situation were met.

4.7.3 Mixed bias

The previous studied situations, despite they allow us to simplify the analysis and understand extreme cases, assume certain hypothesis of independence that are not frequently found in real scenarios. Consequently, the question arises whether the conclusions derived above – relevance probability as optimal true criteria and optimality of average rating and relevant popularity in true and observed precision, respectively – are extensible to any other situation in which the independence conditions required in previous cases do not take place.

The answer is no. In fact, with no further assumption than the relevance-neutral user rating behaviour, as depicted in Figure 4.6.b, the optimal precision rankings get defined by:

$$f(k) \sim \frac{1}{1 + \frac{1 - a\,p(seen|\neg rel, R_k)}{1 - a\,p(seen|rel, R_k)} \frac{1 - p(rel|R_k)}{p(rel|R_k)}} \propto \frac{1 - a\,p(seen|rel, R_k)}{1 - a\,p(seen|\neg rel, R_k)} \frac{p(rel|R_k)}{\left(1 - p(rel|R_k)\right)} \tag{4.13}$$

$$\hat{f}(k) \propto \frac{p(seen, rel|R_k)}{1 - a\,p(seen|R_k)} \tag{4.14}$$

where $a = \rho\, p(rated|seen)$. The last step of Equation 4.13 holds because $g(x) = 1 \,/\, (1 + x)$ is a decreasing function of x.

Depending on the interplay of discovery and relevance distributions, any result is therefore possible. We can however deduce some general behaviours from the observation of the previous formulas.

In terms of true precision, the relevance probability is not any more the only determinant factor, since $f(k)$ also increases with $p(seen|\neg rel, R_k)$ and decreases with $p(seen|rel, R_k)$. Namely, it is better to recommend items that have mainly discovered by users who do not like them than by users who like them. This makes sense, since already discovered items are excluded from the recommendation, so the larger the number of non-relevant excluded (i.e. discovered) items, the greater the number of mistakes avoided to the recommendation algorithm. Moreover, recommending items that have already been discovered by most of their potential "relevant" users leaves only the users who consider them non-relevant as candidates to receive the recommendation. Such previous behaviour agrees with the general notion that recommending becomes more useful when the discovery processes fails, namely, when non-relevant items are discovered to a larger extent than relevant ones. Despite the influence of previous discovery terms, it must be remarked that $f(k)$ still increases with $p(rel|R_k)$, since $p(rel|R_k) \,/\, \left(1 - p(rel|R_k)\right)$ is a increasing function of $p(rel|R_k)$. Actually, such increase is more than lineal, so in general we can expect a stronger influence of the relevance distribution than of the discovery one.

Regarding observed precision (Equation 4.14), it increases with both $p(seen, rel|R_k)$ and $p(seen|R_k)$ – and thus with $p(seen|rel, R_k)$ and $p(seen|\neg rel, R_k)$. It seems therefore that for observed precision items highly discovered are preferable, especially if they are relevant. This makes sense, since most discovered items will have more (relevant) ratings and, consequently, more test ratings, because the number of total, test and train ratings correlates when we randomly split the data. This contradiction with true precision may lead to situations where observed precision gives quite the wrong impression.

From Equation 4.14 we can deduce another important aspect of the observed precision behaviour: the influence of the discovery distribution bias. It is not just that recommending most discovered items will obtain the best observed results, but also the higher the bias of the discovery distributions $p(seen, rel|R_k)$ and $p(seen|R_k)$ (i.e. the higher the difference in the discovery level of different items), the larger the difference $\hat{f}(k) - \hat{f}(k+1)$ for the items R_k and R_{k+1} of the optimal observed ranking, and thus the bigger the loss in the expected observed precision when swapping both items (see demonstration of Lemma 1). In other words, an increase of the discovery distribution bias will in turn increase the maximum expected precision that is possible to reach by a non-personalized algorithm, since it increases the difference between optimal recommendation and any other ranking.

We have seen, thus, that the bias of $p(seen, rel|R_k)$ and $p(seen|R_k)$ clearly favours maximum observed precision, whereas the effect in terms of true precision is not so clear, because it is mitigated by the relevance distribution.

Now let see what we can deduce about the behaviour of the popularity-based recommendations. The ranking functions for both total and relevant popularities are the following:

$$pop(R_k) \propto p(seen|R_k)$$

$$rpop(R_k) \propto p(seen|rel, R_k)p(rel|R_k)$$

The optimal observed criterion $\hat{f}(k)$ is therefore an increasing function of both popularities – it can be rewritten as $\hat{f}(k) \propto rpop(R_k) \,/\, \big(1 - a\, pop(R_k)\big)$ – so we can expect they present a good behaviour in observed precision, unless the popularity distributions are too flat or work against each other, as in some cases we have characterized in previous sections. In terms of true precision, however, the situation is much more unpredictable. Relevant popularity will favour at the same time items with high relevance probability $p(rel|R_k)$ but also with high discovery recall $p(seen|rel, R_k)$, which works against true precision. Total popularity does not even take into account user's tastes, which can lead it to random recommendation or worse, as we see in previous sections.

Regarding average rating we have:

$$avg(R_k) \sim \frac{p(seen, rel|R_k)}{p(seen|R_k)} = \frac{1}{1 + p(seen, \neg rel|R_k) \,/\, p(seen, rel|R_k)}$$
$$\propto \frac{p(seen|rel, R_k)}{p(seen|\neg rel, R_k)} \frac{p(rel|R_k)}{\big(1 - p(rel|R_k)\big)}$$

In the same way that relevant popularity, the term $p(seen|rel, R_k)$ works against true precision, and in this case so does the term $p(seen|\neg rel, R_k)$. With two terms contrary to true precision instead of one, we could think that average rating is just worse than

Description	**Section**	**Subcases**	**Label**	
Relevance-independent rating bias $rated \perp rel \mid i$	4.4	$p(rel\|i)$ steeper enough than $p(rated\|i)$	a	1
		$p(rated\|i)$ steeper enough than $p(rel\|i)$		2
		Neither dominates		3

Table 4.1. Description (at rating distribution level) of the different situations that can potentially take place with a relevance-independent rating bias. Combine this table with Table 4.3, via the column Label of both tables, to obtain the (observed and true) performance of each recommender in each situation.

Focus bias (and neutral assumption in the other bias)	**Section**	**Cases / simplifying assumptions**	**Subcases**	**Label**	
Rating decision bias $seen \perp rel, i$	4.6	Item independence $rated \perp i \mid seen, rel$	$p(rated\|seen, rel) > p(rated\|seen)$	e	1
			$p(rted\|seen, rel) < p(rated\|seen)$		2
Discovery bias $rated \perp rel, i \mid seen$	4.7.1	Relevance bias $seen \perp i \mid rel$	$p(rel\|seen) > p(rel)$	c	1
			$p(rel\|seen) < p(rel)$		2
	4.7.2	Item bias $seen \perp rel \mid i$	$p(rel\|i)$ steeper enough than $p(seen\|i)$	d	1
			$p(seen\|i)$ steeper enough than $p(rel\|i)$		2
			Neither dominates		3

Table 4.2. Description (at discovery distribution level) of both rating decision bias and discovery bias. Combine this table with Table 4.3, via the column Label of both tables, to obtain the (observed and true) performance of each recommender in each situation.

relevant popularity. But it must be noted that an extra dependency from relevance is included via the term $\left(1 - p(rel|R_k)\right)$ in the denominator, which draws it closer to optimal true ranking. As we mentioned before, the increase of the term $p(rel|R_k) \,/\, \left(1 - p(rel|R_k)\right)$ with respect to $p(rel|R_k)$ is more than lineal, so we may generally expect that such term will dominate the product in both average rating and true optimal ranking, and therefore average rating will present a better general behaviour in terms of true precision. However, a too flat relevance distribution, compared with the discovery one, can cancel such relevance tendency of both average rating and true optimal ranking, leaving discovery effects make them take opposite directions.

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">Total popularity</th><th colspan="2">Relevant popularity</th><th colspan="2">Average rating</th></tr>
<tr><th colspan="2">Label</th><th>$\mathbb{E}[\widehat{\boldsymbol{P}}@\mathbf{1}]$</th><th>$\mathbb{E}[\boldsymbol{P}@\mathbf{1}]$</th><th>$\mathbb{E}[\widehat{\boldsymbol{P}}@\mathbf{1}]$</th><th>$\mathbb{E}[\boldsymbol{P}@\mathbf{1}]$</th><th>$\mathbb{E}[\widehat{\boldsymbol{P}}@\mathbf{1}]$</th><th>$\mathbb{E}[\boldsymbol{P}@\mathbf{1}]$</th></tr>
<tr><td rowspan="3">a</td><td>1</td><td colspan="2" rowspan="3"></td><td rowspan="6" colspan="2"></td><td colspan="2" rowspan="3"></td></tr>
<tr><td>2</td><td>Optimal</td><td rowspan="2">Random</td><td colspan="2">Random</td></tr>
<tr><td>3</td><td>> Random</td><td colspan="2">> Random</td></tr>
<tr><td rowspan="2">e</td><td>1</td><td colspan="2">Optimal</td><td rowspan="6">Optimal</td><td rowspan="6">Optimal</td></tr>
<tr><td>2</td><td colspan="2">Worst</td></tr>
<tr><td rowspan="2">c</td><td>1</td><td colspan="2">Optimal</td></tr>
<tr><td>2</td><td colspan="2">Worst</td></tr>
<tr><td rowspan="3">d</td><td>1</td><td colspan="2" rowspan="3"></td><td rowspan="3" colspan="2"></td></tr>
<tr><td>2</td><td>Optimal</td><td rowspan="2">Random</td><td colspan="2">Random</td></tr>
<tr><td>3</td><td>> Random</td><td colspan="2">> Random</td></tr>
</table>

Table 4.3. Observed and true expected precisions of each of the popularity-based variants in each of the situations described on Tables 4.1 and 4.2.

Label	$\boldsymbol{f(k)}$	$\boldsymbol{\hat{f}(k)}$	$\boldsymbol{rpop(R_k)}$	$\boldsymbol{avg(R_k)}$	$\boldsymbol{pop(R_k)}$	$a-b$
a	$p(rel\|R_k)$	$p(rel\|R_k)p(rated\|R_k)$	$p(rel\|R_k)$	$p(rel\|R_k)$	$p(rated\|R_k)$	-
e					$(a-b)p(rel\|R_k)+b$	$p(rated\|seen,rel) - p(rated\|seen,\neg rel)$
c						$p(seen\|rel) - p(seen\|\neg rel)$
d		$p(rel\|R_k)p(seen\|R_k)$			$p(seen\|R_k)$	-

Table 4.4. Ranking functions of the optimal rankings and the different popularity-based recommenders in each of the situations described on Tables 4.1 and 4.2.

To sum up, we have seen that relevant popularity behaves quite like the optimal observed ranking, which in turn is clearly rewarded (in terms of observed precision) by discovery biases. In terms of true precision, however, such biases do not seem to affect (or not so clearly) the optimal true ranking, which behaves in general more like average rating. Therefore, strong discovery biases may increase the possibilities of contradiction between observed and true results, since in situations with such biases, relevant popularity will obtain a notably higher observed performance than average rating (what we are indeed observing in standard offline experiments), whereas it is the latter which is actually performing best.

Note that we expose here some general trends of what we can expect in a generic scenario, but nothing prevents total popularity, relevant popularity and/or average rating

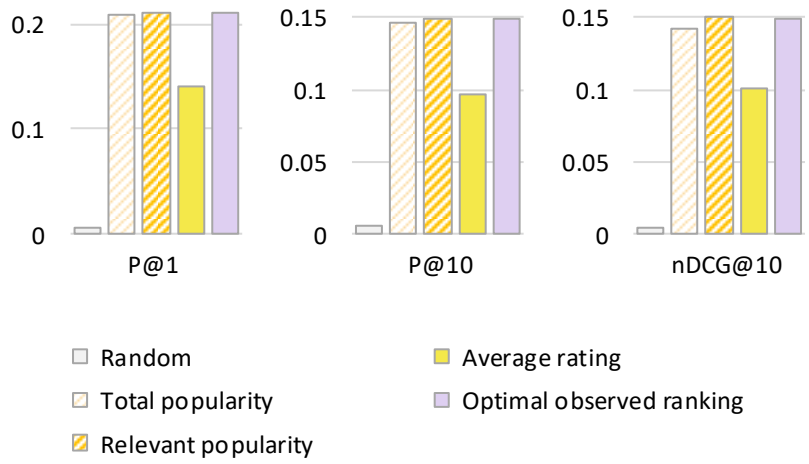

Figure 4.7. Accuracy of each recommender – in terms of $P@1$, $P@10$ and $nDCG@10$ – on MovieLens dataset.

to obtain a true accuracy worse than random. We indeed will see simulated examples of that situation in next chapter.

4.8 Empirical observation

We show and explain now the empirical experiments carried out to validate the theoretical findings obtained in the previous sections. We will use Tables 4.1, 4.2 and 4.3 to refer such different situations and remember the conclusions derived from them.

Most public datasets commonly used in offline evaluation come from portals or applications (MovieLens, Netflix, Last.fm, etc) where users voluntarily and spontaneously manifest their opinions about items they have previously discovered. Such discovery takes usually place, at least partly, out of the application and it is therefore impossible to know how it is distributed over non-rated items. Moreover, the full relevance distribution is also unknown, and only the relevance manifested by user's ratings is available, i.e. the observed relevance. Using such collections, we can therefore only compute observed metrics, and we have no means to contrast them with true values or to know in which of the studied situations we are.

Figure 4.7 shows an example of this usual situation, with the MovieLens dataset. Such figure represents an experiment quite like the one we used in Chapter 2 (Figure 2.2) to contextualize the research carried out in this thesis. Unlike such previous analysis, here and in the next experiments we use the binary version of average rating with an additive smoothing (Zhai & Lafferty 2004). The reason of such smoothing is avoiding the bias that this algorithm presents towards recommending anomalous items with very few ratings. The resulting final formula of $avg(i)$ is therefore given by the following expression:

$$avg(i) = \frac{|i^{+}_{train}| + \mu\, p(rel)}{|i_{train}| + \mu}$$

where $\mu = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}}|j_{train}|$ is the average number of ratings per item and $p(rel)$ is the fraction of ratings that are positive in the dataset. In addition to the effectiveness obtained by the four non-personalized algorithms – random recommendation, both total and relevant popularities and average rating – the precision of the optimal observed ranking is also included in this figure.

We see that total and relevant popularity seem quite the same and pretty equivalent to the optimal ranking, whereas average rating – despite the smoothing – is well below them, very far from being optimal. These results match the general good behaviour we expect from popularities in observed precision, but we cannot explain or verify much further with the available data.

4.8.1 A Crowdsourced Dataset

We therefore need additional relevance information in order to compute true precision. Ideally, all full relevance knowledge would be necessary to compute the exact value of such metric. This is impossible to implement in practice, since we would have to ask each user about her opinion – an opinion probably not formed yet – on each specific item in the dataset. However, an unbiased sample of the true relevance distribution would be enough to obtain a relative algorithm comparison that reflects reality, and such comparison is what we are really interested about in order to contrast our findings.

The strong biased popularity distributions of common public datasets, which we depicted in Section 2.3, may suggest that the relevance we observe on such datasets – and that we use to recommend and evaluate – is probably influenced by the potential biases of the discovery distribution and it is therefore not a representative sample of the true underlying relevance. In order to describe the situation, and characterize the potential biases of the observed results, we will also need – aside the unbiased relevance sample – extra (and also unbiased) information about how discovery is distributed over non-rated items. To the extent of our knowledge there is no dataset with these characteristics. It is certainly true that Yahoo!R3[1] dataset (Marlin & Zemel 2009) contains an unbiased sample of the relevance distribution but it is quite a small one (only 10 ratings per user) and it does not come with any discovery information.

We are therefore looking for a dataset where, first, observed relevance represents an unbiased sample of the true underlying relevance distribution and, secondly, the discovery distribution is known or can be estimated – without any bias interference – from the sample. According to this twin objective, we carry out an experiment with real users

[1] https://webscope.sandbox.yahoo.com/catalog.php?datatype=r

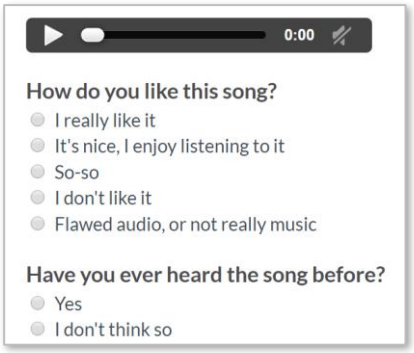

Figure 4.8. Interface of each music track questionnaire.

where, in absence of discovery biases, we obtain their preferences over a series of items and collect information about whether they already knew such items or not.

To avoid discovery biases in the rating sample, we randomly select the items that users must rate and then ask them to do it. Therefore, users might not know the items in advance and have thus to form an opinion just at that moment. In order to obtain the largest number of ratings per user, such opinion formation must be as quick as possible, practically instantaneous. This limits the item domain, since products as films or books are excluded. For this reason, we use the music domain, where a user can judge whether she likes or not a song by listening only a few seconds thereof.

We thus start the experiment by randomly sampling music tracks from a large database, Deezer[2], containing over 30 million songs at the time of this experiment. We did so by randomly generating track IDs in the Deezer range, and pasting the IDs into HTTP requests for obtaining the corresponding 30 seconds music tracks. Following this process, we generate a total of 1,100 audios. The objective was to obtain more than 1,000 valid songs, so we take such margin of one hundred in case some of the downloaded audios could be damaged.

After the music sampling, we ask around 1,000 users of CrowdFlower, a crowdsourcing platform that now has changed its name to Figure Eight[3], to rate some of these songs. Specifically, we randomly assign 100 tracks to each user – each track is thus assigned to about 100 users – adding to a total of around 100,000 assignments. For each assigned song, users must choose one option between the five we offered them: four referred to the opinion of the user about the song and the last one was used to detect flawed or not music audios. We also use this last option to filter out unreliable users, since we intentionally introduce flawed music at a random position every 12 tracks and discard users who fail to properly identify it. From the four possible answers to evaluate the relevance degree,

[2] https://www.deezer.com/

[3] https://www.figure-eight.com/

Nr. users	1,054
Nr. items	1,084
Nr. ratings	103584
Rating density	9.07%
Relevant rating density	2.59%
Nr. users who already knew the song	11,594
Discovery density	1.01%
Relevant discovery density	0.64%

Table 4.5. Volumetric details of CM100k.

Title	Artist	Nr. of relevant ratings
I Will Survive	Gloria Gaynor	87
Fur Elise	Classical Study Music	83
Piano Sonata in A Major	Eliso Bolkvadze	73

Table 4.6. Rating stats of the three most liked songs of CM100k.

two of them present a clear positive nuance ("I really like it" and "I'ts nice, I enjoy listening to it"), other was neutral ("So-so"), and the last one was clearly negative ("I don't like it"). In the experiments of the next sections, we take the top two answers as indicating relevance, and the next two as non-relevance. Along with the question about the user's opinion, we also ask her if she has heard the song before. The objective of this question is to obtain a sample of the discovery distribution.

Figure 4.8 shows the user interface we just describe above. We can see that it includes a player to listen the song. To avoid any external influence or potential bias, no tittle or author information is provided, and only the audio is available to judge the relevance of the song.

Note that 100 ratings per user implies that 90% of the discovery and relevance distributions is still unknown. However, the random assignation between users and items allows us to assume that both obtained distributions are representative samples of the true ones, since we completely remove the discovery bias. Moreover, we completely remove the rating decision bias by requiring users to rate everything they are presented with. Such bias absence, together with the collected discovery information, is a completely novel aspect that distinguishes this dataset from others. Along this thesis we will refer to this crowdsourced dataset as CM100k[4], and its dimensions can be consulted in Table 4.5. In

[4] The dataset is publicly available at http://ir.ii.uam.es/cm100k

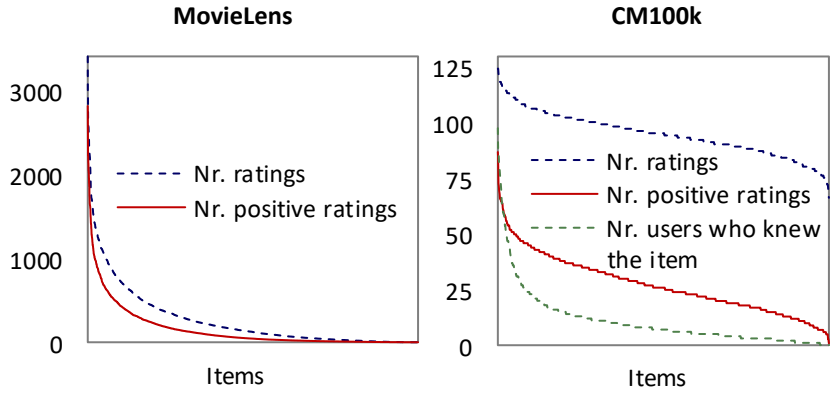

Figure 4.9. Data distributions in MovieLens and CM100k. Each point in every graph corresponds to a music track in the dataset. Note that each curve has axis x (items) sorted by decreasing order of the corresponding distribution. The x values of the curves therefore do not match with each other.

Table 4.6 we present the information about the three songs that present a higher number of relevant ratings.

Figure 4.9 shows the total and relevant rating distributions of CM100k vs. MovieLens. For the former we additionally show the discovery distribution. Note that here and in the next sections the random variable $seen$ takes value 1 if the user already knew the song. We can see that in CM100k the random rating assignment process gives rise to a rating uniform distribution (with a natural binomial variance), whereas discovery and relevance are heavy-headed. The distribution of positive ratings, however, is quite flatter than the one of MovieLens. We will comment the consequences of this difference in the next section.

On the other hand, the scatterplot Figure 4.10.b shows the relation between discovery $p(seen|i)$ and relevance $p(rel|i)$ for each music track. We can see that the most known song corresponds to the most liked ("I Will Survive" of "Gloria Gaynor"), although with less popular items the correlation is loose. This makes sense, and means that there is some degree of accuracy in the discovery processes that give rise to such distribution. In fact, we specifically observe quite a high global discovery precision of $p(rel|seen) = 0.6131$.

4.8.2 Evaluation under different scenarios

Using the CM100k dataset we can recreate most of the different situations studied in Sections 4.4, 4.6 and 4.7. We describe each situation and the corresponding experimental results in turn in the following paragraphs (labelled a-d matching the Figures 4.10, 4.11 and 4.12 and tables 4.1 to 4.4).

In all of them, we randomly divide the rating data into training and test sets with a split ratio of $\rho = 0.8$. Then, we use the training data to run the recommendation algo-

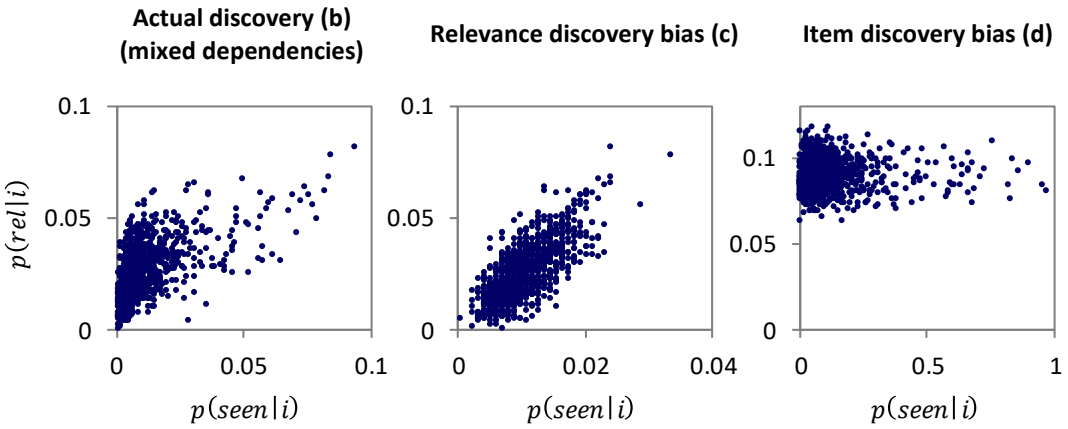

Figure 4.10. Dependency between relevance and discovery in the scenarios b, c and d. Each point in the plots corresponds to an item of the dataset. The x axis shows the fraction of users who know the item – i.e. $p(seen|i)$ – and the y axis represents the users who like it – $p(rel|i)$.

rithms and the test set to compute observed accuracy metrics. If more relevance judgments are available – it depends on the situation – we compute the true values by adding to the test set such extra relevance information.

a) Rating fully independent from relevance and items

Because of the random rating assignment that leads to CM100k dataset, the rating variable does not depend on any other variable in such dataset, particularly it is independent from relevance and items. This fits therefore with the situation described in Section 4.4, where we assumed independence between relevance and rating. Here at the same time we also have item independency, i.e. $p(rated|rel,i) = p(rated)$, so relevance distribution is particularly steeper than rating distribution – the latter is uniform indeed – and we have specifically case a.1 of Table 4.1.

Figure 4.11 shows the results of running the recommendation algorithms on CM100k as is, considering all the available data as observed ratings. As we predicted, both average rating and relevant popularity present quite a good behaviour in terms of observed precision, whereas total popularity is at the same level that random recommendation. We can also see that the advantage of relevant popularity over random is smaller than in MovieLens (Figure 4.7). This happens because the relevant rating distribution is much less steep in CM100k (see Figure 4.9). Note that only observed precision is computable in this case, since we are using all the available judgements as ratings. Moreover, we are ignoring the discovery information about what users already knew.

b) Mixed discovery dependencies

If we now take into account the discovery information, CM100k fits with the mixed scenario described in Section 4.7.3, where discovery depends on both relevance and items. Such dual dependency can be deduced from the plot of Figure 4.10.b: there is a clear

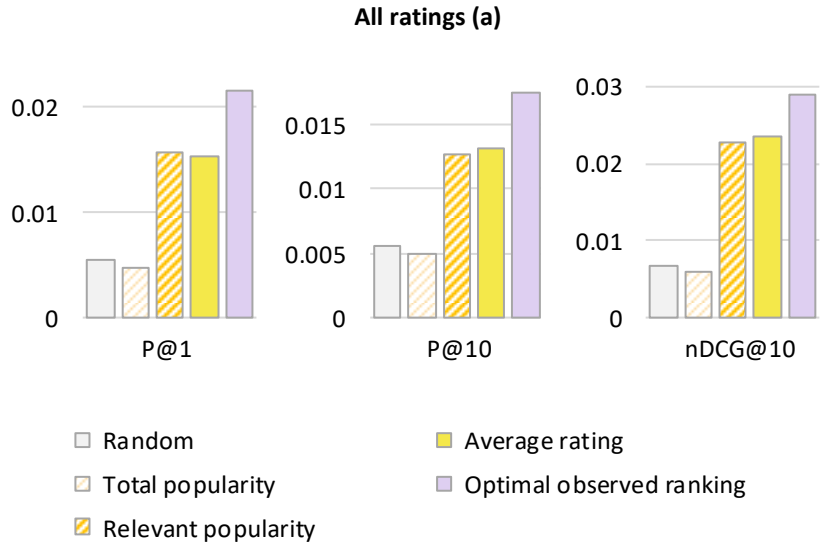

Figure 4.11. Accuracy of each recommender – in terms of $P@1$, $P@10$ and $nDCG@10$ – on CM100k dataset. All ratings are given as input to recommenders (scenario a).

connexion with relevance, but it is not the only factor, since different items present distinct discovery levels. The independence hypothesis referring the rating decision process that is assumed in the theoretical scenario of Section 4.7 – i.e. neutral user behaviour – is trivially met here since we are taking such decision from users by forcing them to rate what we randomly select. It does therefore not depend from any other variable, particularly nor relevance or items: $p(rated|seen, rel, i) = p(rated|seen)$.

Note the difference with previous scenario. There, the rating variable ($rated$) was independent from both relevance and items, but here is only the rating decision ($rated|seen$) which is independent from them. In fact, in this case $rated$ obviously depends on relevance and items, since $seen$ does.

In order to empirically recreate such scenario, we consider as ratings only the opinions for music that users declared to know. These opinions are the only ones we could have obtained in a common real scenario, where users rate what they know. We therefore provide as input to the recommendation algorithms an 80% (training set) of such opinions on already known items, leaving the other 20% (test set) to compute observed metrics. The rest of the (non-discovered) judgements are using, along with the test set, to compute true metric values.

Figure 4.12.b shows the results obtained when evaluating recommendation algorithms with the previous settings. We can see that observed precision behaves quite similar than in MovieLens, with both popularities close to the optimal recommendation and average rating below of them. True precision however tells quite a different story, revealing that total and relevant popularities are indeed slightly below random. Average rating, on the contrary, do not present such a poor behaviour, but is still not far from random recommendation.

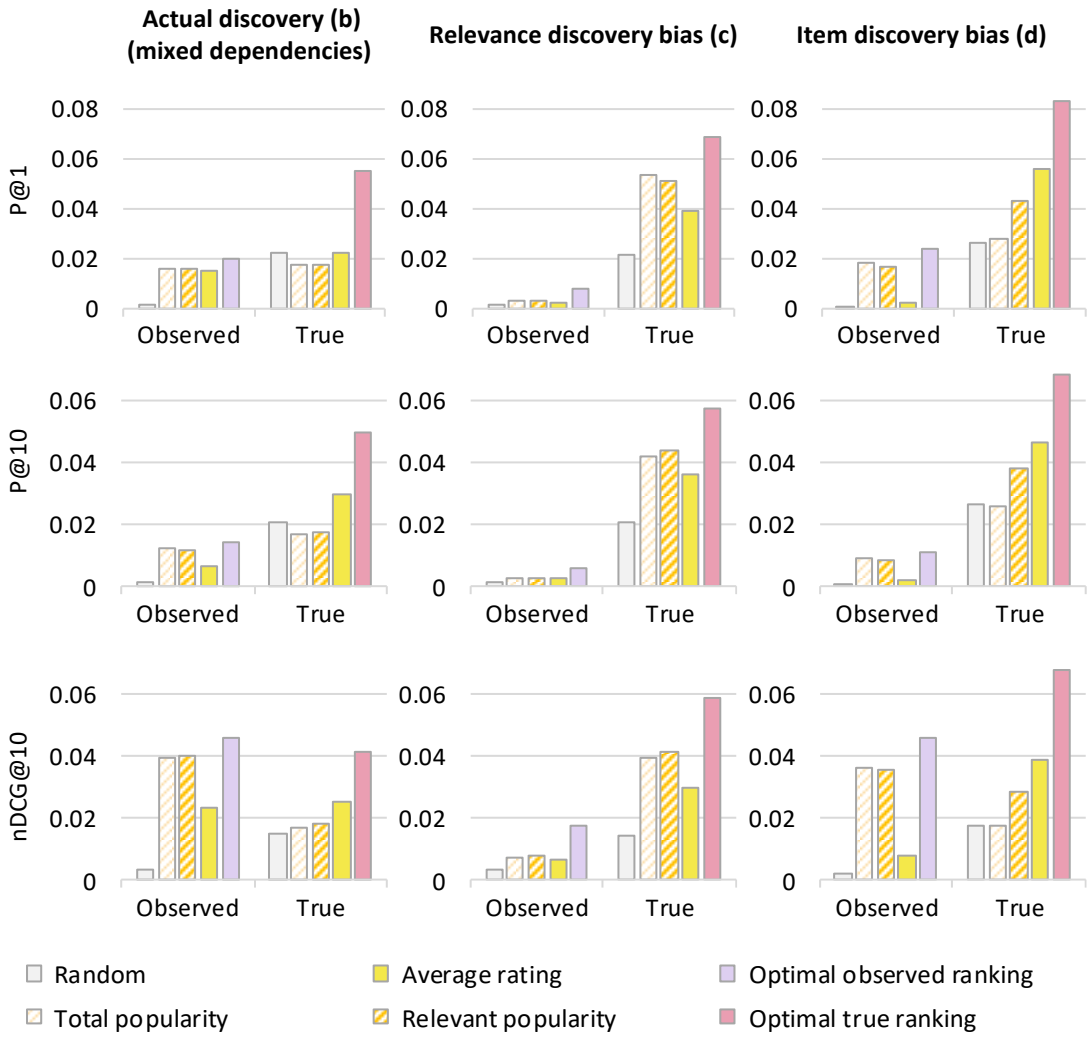

Figure 4.12. Observed and true accuracy values – in terms of $P@1$, $P@10$ and $nDCG@10$ – that both optimal rankings and popularity-based recommenders obtain in scenarios b, c and d.

When we previously analyse this situation in Section 4.7.3, a situation where discovery depends both on relevance and items, we concluded that it can gives rises to any potential outcome. We deduced some general trends that actually match the results we obtain here – popularity is good in observed precision but average rating is better in true precision – however, there is no reason to assume we may grant such outcome. Other relevance and discovery distributions could lead to a different result.

The most remarkable conclusion arising from this scenario is that a contradiction between observed and true precision is indeed plausible, and that we should take care about such possibility when evaluating recommendation algorithms according to offline methodologies.

c) Relevance discovery bias

Given that the collected data in CM100k presents both relevance and item discovery biases, in order to recreate separately scenarios with only one of these biases, we shall reassign somehow the discovery along the judgments. In this section, we take care of the relevance bias, postponing the recreation of a scenario with just item-dependent discovery for the next point.

To remove the dependency between items and discovery – and leave therefore only the relevance dependence – we reassign the discovery of each user-item pair according to the global probability $p(seen|rel)$ if the item is relevant for the user and to $p(seen|\neg rel)$ otherwise. Both values $p(seen|rel)$ and $p(seen|\neg rel)$ are estimated from initial data by counting the ratio of relevant (or non-relevant) items that were already known. Note that we are considering as non-relevant the opinions we did not collect, so for each of them we should also assign $seen = 1$ (and thus add it to the rating data taking value $rel = 0$) with probability $p(seen|\neg rel)$. By doing the previous modifications, we recreate the scenario studied in Section 4.7.1. In Figure 4.10.c we can observe how the resulting $p(seen|i)$ distribution is almost proportional to $p(rel|i)$, proving that with the previous reassigning process we have removed item dependency from discovery. Moreover, since in our data we have $p(seen|rel) = 0.2473 > 0.0038 = p(seen|\neg rel)$, the setting specifically matches case c.1 in Table 4.2.

As in the previous setting b, to run the different recommendation algorithms we only consider as ratings the opinions for which $seen = 1$. Ratings that we then randomly split in order to compute both observed and true precisions. Figure 4.12.c shows the corresponding empirical results, that are indeed consistent with the analytical ones: both popularity variants and average rating are near-optimal in both observed and true precision.

d) Item discovery bias

For reproducing a scenario where discovery only depends on items, we shall therefore remove the dependency of such discovery from relevance. In order to do that, we first randomly shuffle the discovery distribution over items, i.e. we reassign each $p(seen|i)$ to a random item j. Then, for each user $u \in \mathcal{U}$, we assign $seen = 1$ to the pair u, j with probability $p(seen|j)$ – probability randomly taken from other item i according to the previous step – and $seen = 0$ with probability $1 - p(seen|j)$. This way, any potential dependency that there might be between $seen$ and rel – given the item – is removed, as we can confirm in Figure 4.10.d. This situation corresponds to scenario d in Table 4.2.

Consulting such Table 4.2, we note that this scenario is in turn divided in three possible cases, regarding the comparison between the steepness of relevance and discovery distributions. The discovery shuffle described above does however not modify neither of these two distributions, so the situation is still the same as depicted in Figure 4.9: neither dominates the product of $p(seen|i)p(rel|i)$ and therefore case d.3 applies.

Repeating the evaluation approach carried out in cases b and c above, we see once again that the results – depicted in Figure 4.12.d – match the analytical prediction. Thus, in terms of true accuracy average rating tends towards the optimal, while relevant popularity is not so close, despite it is still clearly better than random. In contrast, observed precision shows the opposite situation: relevant popularity is close to optimal, and average rating is just better than random. The contradiction becomes more remarkable in the case of total popularity, which is near optimal in observed accuracy, but random-level in terms of true accuracy.

As we predicted in the theoretical analysis, average rating seems a more reliable option than relevant popularity in terms of true precision. In fact, the optimality of average rating is granted in this scenario without any additional condition, whereas the performance of relevant popularity depends on the steepness condition.

We have therefore checked that the empirical results agree with the theoretical ones in all the studied situations. We see indeed that, despite that the analytical development is focus on $P@1$ due to is tractability, it generalizes empirically well to deeper cutoffs ($P@10$) and other and more complex accuracy metrics ($nDCG@10$).

4.8.3 Personalized algorithms

In this chapter we have focused on how experimental methodologies can distort the observed effectiveness of popularity-based recommendations, giving rise to a completely misleading message. Someone can argue about the utility of paying so much attention to a non-personalized algorithm whose accuracy is obviously below than that of other more complex and personalized methods.

Some of such methods have however been proved to be somehow determined by popularity, as we introduced in Chapter 2, so any of the distortions characterized in this chapter can potentially affect them. In order to show that this is not an artificial and implausible problem, we compare here two variants of the user-based k nearest neighbours (kNN) algorithm: normalized and non-normalized. We will prove in Chapter 6 that the normalized variant is biased towards average rating while the non-normalized one is influenced by relevant popularity. According with such trends, there is the possibility that each of these versions follows the behaviour of its corresponding non-personalized reference, giving rise to a potential contradiction in certain situations.

Figure 4.13 shows the performance of the two kNN versions in the most general, mixed dependency scenario (case b), as well as in the MovieLens dataset. We take $k = 10$ and $k = 80$, for the normalized and non-normalized variants respectively, when running on MovieLens, while for CM100k we just take all users as neighbours. In the normalized version we also require a minimum of 3 neighbour ratings for an item in order to be recommended. We can see that the (popularity-biased) non-normalized variant performs much better than the normalized one in terms of observed precision, both on MovieLens

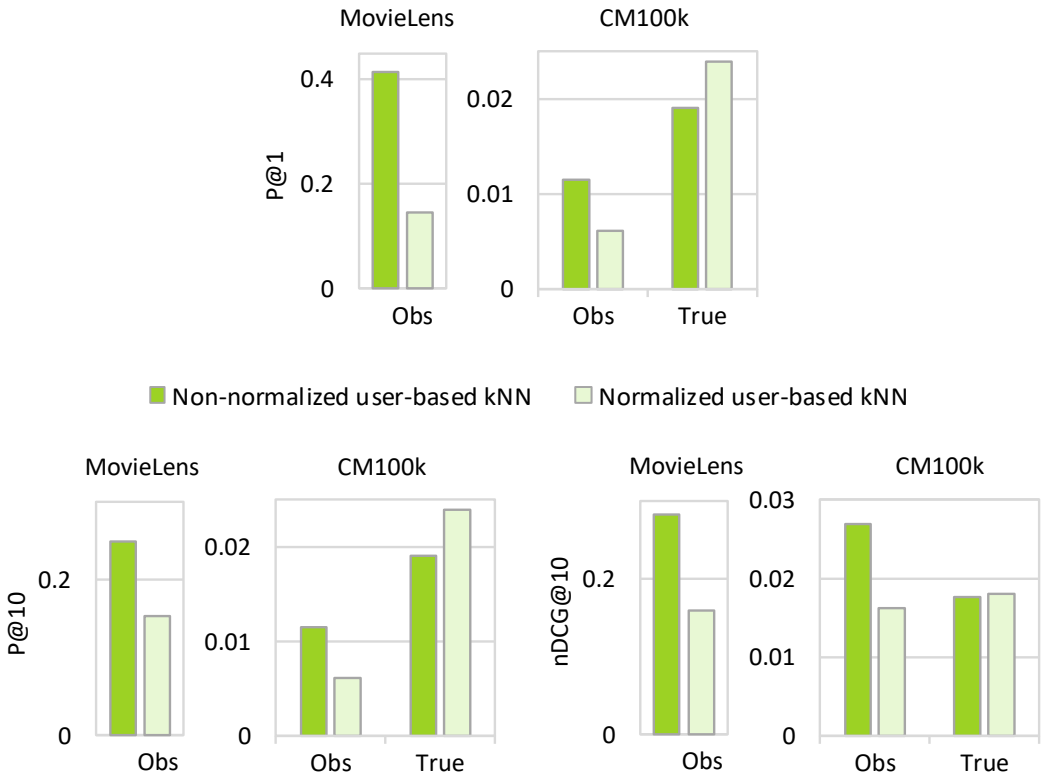

Figure 4.13. Accuracy of kNN (normalized and non-normalized variant) – in terms of $P@1$, $P@10$ and $nDCG@10$ – on CM100k and MovieLens dataset.

and CM100k. Note that this is also the situation in Netflix and Last.fm, as we depicted in Chapter 2, Figure 2.2. However, the true values reveal just the opposite situation, claiming that the algorithm biased towards average rating is indeed preferred. One may wonder if we would see a similar result in MovieLens, Netflix or Last.fm if we had an unbiased relevance sample.

4.9 Conclusions

In this chapter, we have provided a theoretical probabilistic framework that allows to describe and analyse the recommendation task in a formal way. Here we point out the main conclusions that arise from this analysis:

- There is a formal explanation of why popularity presents such a good performance in common offline experiments. First, we see that popularity is indeed rewarded by observed precision, presenting observed values quite close to the maximum possible and secondly, that such maximum increases with the popularity biases. In other words, the more unequal rating distribution, the bigger the maximum accuracy that

can be observed in a non-personalized recommendation and, since popularity is close to such maximum, the bigger observed performance it will obtain.

- Relevant popularity is quite a better option than total popularity. It is more robust, for instance, against atypical user behaviours or discovery biases that run against relevance. However, average rating results in a more reliable option than both, presenting optimal behaviours even with strong item dependency biases. Even so, all these popularity variants could potentially behave worse than random is scenarios with mixed dependencies.
- The values measured in common offline experiments can strongly disagree with the true ones, even in the relative comparison of two or more algorithms. That is, with such experiments we might wrongly choose the best algorithm amongst several options. The main factors that can lead to this erroneous behaviour are the discovery biases that are not only based on user's tastes, but also depend on other particularities of the items. Moreover, the bigger the bias of the discovery distribution over different items, the larger the likelihood of this contradictions between observed and true results.
- We have also shown that such potential contradictions can not only affect popularity-based recommendations, but they may indeed misrepresent the comparison of other personalized and more elaborated methods.
- The best possible non-personalized recommendation, in terms of true accuracy, is obtained by sorting the items according to the fraction of non-observed opinions that are relevant. This implies a revision of the Probability Ranking Principle (PRP) to consider the particularities of recommender systems, specifically the exclusion of already rated items from the delivered ranking.

The generation of a novel dataset is another important finding of this chapter. The absence of any kind of discovery or rating decision bias draws an important distinction between CM100k and the rest of datasets in the area. It particularly allows to recreate a standard offline experiment, in which extra relevance information is available to unbiasedly compute true accuracy metrics.

Chapter 5

Popularity biases derived from social network dynamics

In the previous chapter we formally analyzed the effectiveness of popularity in recommendation in different characteristic situations. The clearest conclusions were reached for prototypical cases involving independence assumptions, without which we explain that any outcome is possible. Seeking further understanding of the general assumption-free case, we now study a particular case where item discovery is mainly a consequence of word-of-mouth in a social network. We thus analyze the interactions between discovery, relevance and ranting that arise from individual user sharing behavior. We carry out such analysis under an empirical approach, by simulating the discovery of new items and the rating generation in a social network environment. The structure of the chapter follows the structure of the publication Cañamares and Castells (2014).

5.1 A social rating generation model

In Chapter 4 we introduced the fundamental actions, events and variables involved in the rating generation process, upon which we carried out the formal analysis of the popularity-based recommendation effectiveness. In this section, we take on the same model but incorporate some further actions related with social interactions. With the new actions, we will be able to simulate network processes and observe the derived effects on the effectiveness of popularity in recommendation.

In the analysis of Chapter 4, we identified two necessary steps (and potential sources of biases) for a user to rate and item: discover it and decide to rate it. Now, we add one more action related with the social environment: the user shares her information about the item with a friend (a contact in the social network). Note that, as a consequence of this action, the recipient of the communication indeed discovers the item.

Thus, these three actions or steps (discovery, rating and communication) create a cycle by which the information about different items progressively traverses the social network: items become known to the users they come across, who might decide to rate

them and to talk about them to other users, who in turn will decide whether rating and sharing them. How far and how quickly an item spreads in the social network depends on the communication patterns of users (for instance, the dependency of the sharing decision with the characteristics of the item or the user's tastes) and the shape and connectivity of the network. In fact, the latter has been proved to may significantly affect the propagation phenomena (Doer et al. 2012).

5.1.1 Random variables and parameters

In Chapter 4 (Section 4.1.1), we already formally model the discovery and rating decision processes in terms of three binary random variables defined upon the sample space $\mathcal{U} \times \mathcal{I}$, where $\mathcal{U}$ is the set of all users and $\mathcal{I}$ the set of all items. Such variables were: $rated: \mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$, defined as $rated(u,i) = 1$ if user u has rated item i and 0 otherwise; $rel: \mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$ that takes value 1 if the user likes the item, and 0 otherwise; and $seen: \mathcal{U} \times \mathcal{I} \rightarrow \{0,1\}$, that is 1 if the user is aware the item exists, and 0 otherwise.

In order to formally introduce the communication decision in the probabilistic model, we shall add now one more random variable: $tell: \mathcal{U} \times \mathcal{I} \times \mathcal{U} \rightarrow \{0,1\}$, defined as $tell(u,i,v) = 1$ if the user u tells the given friend v about the item i when both friends talk to each other, and 0 otherwise. Note that we are assuming that users only share information with people they are connected in the social network. This does not imply any loss of generality, since we do not make any assumption on the structure of the network at this point. The simplifying restriction will be made in our experiments, where we will use or simulate specific social networks.

As in Chapter 4, the relevant factors of this social rating generation model can be expressed in terms of conditional probabilities using previous defined random variables. Specifically, we focus on modelling the user (rating and sharing) decisions, in order to observe how the potential biases on these decisions may affect to both the resulting rating data distribution and the effectiveness of recommendation algorithms when using such rating data. Thus, we can express the propensity of users to rate items they like vs. items they do not (i.e. the rating decision) with the conditional probabilities $p(rated|seen, rel, i, u)$ and $p(rated|seen, \neg rel, i, u)$, respectively, as we did in Chapter 4. Here, in addition, we can model the user inclination to share positive vs. negative experiences (sharing decision) with $p(tell|seen, rel, u, i, v)$ and $p(tell|seen, \neg rel, u, i, v)$.

We now make the simplifying assumption that previous decisions mainly depend on the relevance of the item, and that the differences that arise from specific users and items can be ignored. In other words, we approximate $p(rated|seen, rel, i, u)$ by $p(rated|seen, rel)$ and $p(tell|seen, rel, u, i, v)$ by $p(tell|seen, rel)$ – and the same for the corresponding non-relevant item condition probabilities.

Thus, we arise to four probabilities that may act as configuration parameters of our model and that allow us to define two potential behavioural biases:

- **Communication bias:** defined by $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$.
- **Rating bias:** defined by $p(rated|seen, rel)$ and $p(rated|seen, \neg rel)$.

In order to empirically observe how these parameters affect the effectiveness of popularity-based recommendation, we shall simulate the previous proposed model. Then, with the rating data resulting of the simulation process we will run the recommendation algorithms and evaluate their effectiveness. The previous model defines a series of actions (discovery, rating and communication) – which in turn implicitly implies the definition of three item-user pair states (unknown, discovered, rated) – but in order to simulate the rating generation process we need to define a set of dynamics, triggering actions and events, and the order in which they take place.

5.1.2 Model dynamics

We now propose a series of dynamics or ordered actions in other to simulate how user-item pairs cross the three potential states defined in the previous model: unknown, discovered and rated (in this order). Note that each state needs for the previous one in order to take place – for instance, a user cannot rate an item without having discovered it before – and that the two latter (discovered and rated) may or may not ever be reached.

Thus, we consider a simulation where users take actions in turns, one after the other. The turn is assigned in a random sweep of the user set, and the unit time consists on an entire iteration over all users. On her turn, each user undertakes the following actions, in the order they are listed:

1. Exogenous discovery: the user discovers (or not) a certain number of items by discovery sources external to the social network.
2. Rating: for each discovered item, the user decides whether introduce a rating for this item or not.
3. Communication: after the rating decision process, and regardless of its result, the user decides whether tell her friends about her new discovered items or not.

We explain and motivate below each of these actions, by indicating how they are simulated and where the model parameters defined in Section 5.1.1 intervene.

Exogenous discovery

Our aim in this chapter is to focus on those discovery biases caused by social network communication. However, if initially all items are unknown to all users, in order to bootstrap the system we shall include an additional discovery source, external to the social network, through which items may also become known. Examples of this kind of source are, for instance, search engines, recommender systems, item advertisements, etc.

This exogenous source of item discovery can be implemented in many ways: as a random sampling in the item space, or as a biased sampling by some arbitrary distribution,

or even as a recommender system. The two latter might significantly alter the resulting rating distribution, so in order to observed only the biases derived from communication effects we choose the random sampling implementation. In other words, items are all equally likely to be discovered by this source.

On each user turn, the number of items to be discovered by external discovery is modelled by a Poisson distribution whose mean λ is a parameter of the simulation. Poisson distribution expresses, from an average frequency of occurrence λ, the probability of a given number of events occurring in a fixed interval of time. In our case, events are discoveries and the interval of time is a user turn.

Rating

The decision to rate already discovered items might take place in different ways and orders. As simplification, we assume that it is only made once, so if the user does not rate the item, the decision is not reconsidered anymore.

In the simulation process, this means that the rating decision is applied only for recent discovery items. After the exogenous discovery takes place, the current user may have discovered certain number of items since her last turn. Some of them may came precisely from such discovery step, but others might be told by other users in their corresponding communication steps. All such recent discovered items are only considered for rating in the current turn, and not in the next ones.

The rating decision is taken based on the probabilistic model described in previous section. That is, for each recently discovered item, the user will rate it with probability $p(rated|seen, rel)$ if she likes the item, and with probability $p(rated|seen, \neg rel)$ if she does not. If the rating finally takes place, it will have the corresponding relevance value: positive if it is relevant and negative if it is not.

Sharing information

Regarding communication, we simulate the decision to share item information following quite the same order and assumptions as in the rating decision process. Namely, on her turn, each user is given a chance to talk about each of her recent discovered items (those items discovered since her last previous turn). Each item may be communicated to a potentially different friend who is sampled uniformly at random from all the user's social contacts.

After the friend selection, the decision to talk or not to such friend about an item is taken according to $p(tell|seen, rel)$ if the user likes the item, and $p(tell|seen, \neg rel)$ if she does not. If communication takes place, the friend discovers the item (if she had not done it yet), so she will be able to rate it and/or share it in her next turn. We simulate communication as a dialog where users tell and ask at the same time, which means that, in the current user turn and if she has decided to talk about an item to a friend, such friend will in turn choose (uniformly at random) some discovered item and (under the same

Algorithm 5.1: Simulation process

Input: $\mathcal{U}, \mathcal{I}$ Sets of users and items, respectively

G Graph with social connections. $G[u]$ with $u \in \mathcal{U}$ denotes the set of users connected with (friends of) u

T Total number of ratings to generate

λ Average number of discovered items (by exogenous discovery) per user turn

$p(rated|seen, rel)$, $p(rated|seen, \neg rel)$ Probabilities guiding rating decision when the item is relevant or non-relevant, respectively.

$p(tell|seen, rel)$, $p(tell|seen, \neg rel)$ Probabilities guiding sharing decision when the item is relevant or non-relevant, respectively

Output: R Rating set

$S \leftarrow \langle (u_1, \emptyset), \ldots, (u_m, \emptyset) \rangle$ // Set of discovered items per user

$D \leftarrow \langle (u_1, \emptyset), \ldots, (u_m, \emptyset) \rangle$ // Set of recent discovered items per user

$R \leftarrow \emptyset$ // No rating data at the starting point

while $|R| < T$ do

for $u \in \mathcal{U}$ do

Exogenous discovery:
- Sample $k \in \mathbb{N}$ with a Poisson distribution with average λ
- Sample $E \subset \mathcal{I} \setminus S[u]$ uniformly at random with $|E| = k$
- $D[u] \leftarrow D[u] \cup E$
- $S[u] \leftarrow S[u] \cup E$

Rating decision:
- for $i \in D[u]$ do
  - if u likes i then $\pi \leftarrow p(rated|seen, rel)$
  - else $\pi \leftarrow p(rated|seen, \neg rel)$
  - with probability π do $R \leftarrow R \cup \{u, i, rel(u, i)\}$

Sharing decision:
- $D' \leftarrow \emptyset$ // Set of discovered items when asking friends
- for $i \in D[u]$ do

Tell:
  - if u likes i then $\pi \leftarrow p(tell|seen, rel)$
  - else $\pi \leftarrow p(tell|seen, \neg rel)$
  - with probability π do
    - sample $v \in G[u]$ uniformly at random
    - $D[v] \leftarrow D[v] \cup \{i\}$
    - $S[v] \leftarrow S[v] \cup \{i\}$

Ask:
  - sample $j \in S[v]$ uniformly at random
  - if v likes j then $\pi \leftarrow p(tell|seen, rel)$
  - else $\pi \leftarrow p(tell|seen, \neg rel)$
  - with probability π do
    - $D' \leftarrow D' \cup \{j\}$
    - $S[u] \leftarrow S[u] \cup \{j\}$
- $D[u] \leftarrow D'$

relevance-based communication probability pattern) talk back about it to the first user, who will then discover this item. Note that under this configuration, users talk about an item on their own initiative only once at most (since they do it only for recent discovered items), but they can talk about it any number of times when asked (this time the item is chosen between all discovered items).

All the previous simulation processes and the order in which they take place are outlined in Algorithm 5.1.

5.2 Simulation setup

The algorithm proposed above allows us to generate different rating datasets depending on the user behaviour patterns that we choose to model and analyse. User behaviour determined by the four conditional probabilities that guide rating and sharing decisions and that work as configuration parameters of our model. In order to analyse the effect that such configurations produce on the effectiveness of popularity-based recommendation, we implement a simulation framework that runs the model dynamics described in Algorithm 5.1.

In addition to the model dynamics, the framework also includes the execution of some recommendation algorithms over the resulting rating data. It indeed supports the integration of as many algorithms as we want. At each simulation time step – i.e. an iteration over all users – the framework generates a temporal split of rating data with a 0.5/0.5 ratio of training/test data. From such split, it runs all the recommendation algorithms taking training data as input, and evaluates each of them with test data for obtaining observed precision. In order to compute true precision, it uses the underlying relevance distribution, about which we talk later in this section.

Note that this time we are implementing a temporal split instead of a random split, as we did in the experiments and analysis of Chapter 4. The reason is that in the scope of a simulation where ratings are being generated in "real time" – and where we indeed have a timestamp for each rating – it is more natural to split the ratings using such temporal information. It supposes a better recreation of real scenarios, where typically the time is also present and, in fact, past interactions are used to predict the future ones.

With this framework we can thus monitor how the performance of recommendation algorithms evolves along the simulation. According with the objectives of the present thesis, we focus on observing and studying the non-personalized recommendation approaches formally analysed in the previous chapter: ranking by popularity, relevant popularity, average rating and random, as well as the two optimal rankings, for observed and true precision, respectively.

Finally, and before analysing the effect of different user behaviour patterns let us define the default setup for the rest of parameters that Algorithm 5.1 takes as input.

Number of users ($\|\mathcal{U}\|$)	4,039
Number of items ($\|\mathcal{I}\|$)	3,706
Social network (G)	Facebook graph
Total number of ratings to generate (T)	668,848 (~4.5% rating density)
Average number of externally discovered items per turn (λ)	0.001

Table 5.1. Default values for the input parameters of Algorithm 5.1.

Thus, as graph G we use the social network data from Facebook that was made available by J. Leskovec in 2012 (McAuley & Leskovec 2012). It contains 88,234 social connections among 4,039 users. Later, in Section 5.5, we will analyse the effect of using another social network structure, but for now we use the Facebook graph as default value. Regarding the set of items, we take inspiration in the order of scale of MovieLens 1M and consider $|\mathcal{I}| = 3{,}706$ items. For exogenous discovery we take $\lambda = 0.001$, namely, on average users discover an item at random 1 out of every 1,000 user turns. We consider such low value to let the bulk of discovery depend mainly on social communication. Finally, we run all the simulations until we obtain the rating density of MovieLens (~4.5%). Table 5.1 summarizes all these default values. For smoothing the variance effects, we average the results over 10 full executions of the simulation.

5.2.1 Relevance distribution

Most of the actions described in Algorithm 5.1 depend on whether the user likes a given item or not, since it determines the probabilities of her decisions. However, relevance is in general an unobserved variable for the system until a user rates an item. In fact, it is also unknown for the user herself until she discovers the item. In order to deal with this lack of observation, we simulate a relevance distribution by defining for each user-item pair if the user likes the item. This relevance information will remain hidden to the system, in particular to recommender systems, but will be made "visible" to a) the simulated users when they discover new items, and b) the computation of true precision.

Our model does not make any assumption about this relevance distribution, but in our experiments, we use the relevance information of CM100k. Such dataset is explained in detail in the previous chapter, and its relevance distribution (number of users who like each item) can be consulted in Figure 4.9 (Chapter 4, Section 4.8.1). We use this dataset because it is gathered in absence of discovery and rating decision biases, and therefore the resulting relevance distribution is an unbiased sample of the full one.

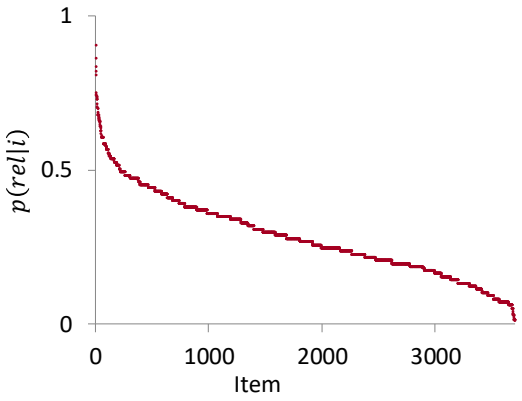

Figure 5.1. Relevance distribution used in the simulation framework. Items are sorted from most to least liked in the x axis, and the line represents the ratio of users who like each item.

We derive the relevance probability of each item by dividing its number of relevant ratings in CM100k by the number of users that have rated it in such dataset. In addition, we scale such probability distribution from the number of items of CM100k ($1{,}084$) to $3{,}706$ by linear interpolation. This way the resulting relevance distribution that we use in all the following experiments is depicted in Figure 5.1. Taking such distribution as input, the simulation framework randomly assigns the different relevance probabilities between items, from which the number of users who like each item is subsequently computed. Then, for each item the specific users that consider it as relevant are selected by randomly sampling.

5.3 Communication bias

To study in isolation the effect of communication biases, we assume a relevance-neutral rating behaviour by taking $p(rated|seen, rel) = p(rated|seen, \neg rel) = 1$, that is, users always rate all items they discover. Then, we shall vary $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$ to reproduce different situations and observe the influence of such parameters in the popularity-based recommendations' performance. In parallel, we will also check the effect of raising the general communication ratio, defined by $p(tell|seen)$.

However, before analysing these sharing biases, and in order to connect the following experimental results with the theoretical conclusions of Chapter 4, let us start by studying how the communication and discovery biases are related.

5.3.1 Discovery bias

Under our assumption of unbiased exogenous discovery, we may intuitively expect that discovery inherits the biases of communication, since the latter is the main cause of the

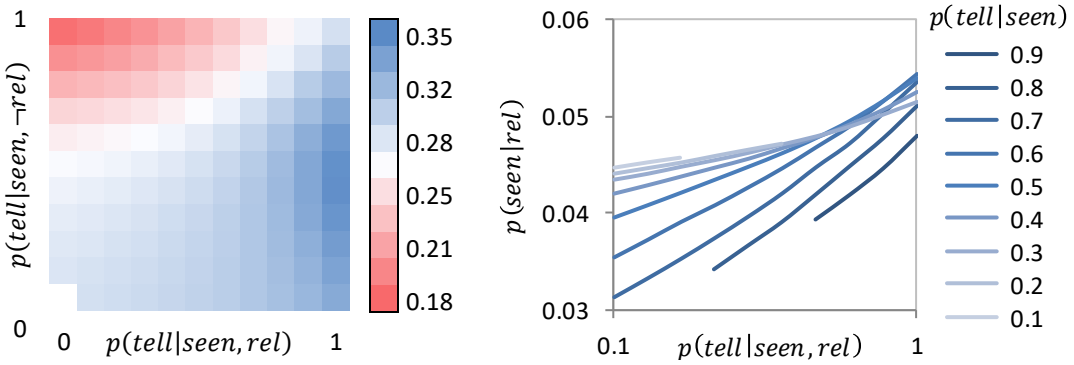

Figure 5.2. Discovery bias – expressed by $p(seen|rel)$ – as function of the communication bias.

former. For instance, in a situation where users talk more about relevant items than about non-relevant ones (i.e. $p(tell|seen, rel) > p(tell|seen, \neg rel)$), we would expect relevant items to be discovered to a larger extent than non-relevant ones ($p(seen|rel) > p(seen|\neg rel)$). This is not necessarily obvious, since the fact that people speak about what they like does not necessarily imply those who listen like it as well.

Figure 5.2 shows how the discovery bias – expressed by $p(seen|rel)$ – varies according to different communication biases. The figure presents the results in two ways: in the colour map on the left, we vary $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$ from 0 to 1 by increments of 0.1, and we show the resulting $p(seen|rel)$ value in a color scale, where blue is the maximum value and red is the minimum. Using this colour map we can verify that the relevance dependence of network communication is almost directly translated to discovery. Thus, the blue zone (high values of $p(seen|rel)$) prevails in the inferior right triangle, i.e. when communication favours relevant items. On the contrary, red cells concentrate in the superior left corner, when users talk mainly about those items they do not like. Note that for this section we are collapsing rating and discovery, so all the configurations finalize with the same number of discovered (rated) items. The discovery prior $p(seen)$ is therefore constant along all the cells in the colormap, so a higher value in $p(seen|rel)$ implies a lower value in $p(seen|\neg rel)$, and we can identify blue zones with situations when relevant items are discovered to a larger extent than non-relevant ones, and red zones with the opposite situation.

The correlation between discovery and communication is however not perfect. For instance, the highest value in $p(seen|rel)$ is obtained in the last column of the map ($p(tell|seen, rel) = 1$) but when $p(tell|seen, \neg rel)$ is around 0.5, not 0. This may suggest that certain level of communication is preferable, even if it implies talking about non-relevant items, in order to allow the communication biases take effect, otherwise the exogenous discovery becomes no negligible and can alter the results.

The right graphic of Figure 5.2 provides a complementary view, where each line corresponds to a value of $p(tell|seen)$ – i.e. the total communication level, both positive and negative –, the x axis is $p(tell|seen, rel)$, and the y axis is the resulting discovery bias $p(seen|rel)$. Note that the curve for $p(tell|seen) = 0.9$ has no values for $p(tell|seen, rel) < 0.6$, since it is not possible to reach such a high prior with lower communication probabilities on relevant items. Likewise, $p(tell|seen) = 0.1$ has no points for $p(tell|seen, rel) > 0.2$, and the same happens with priors 0.2 and 0.8.

Consulting these curves, we confirm the general trend of the colour map: $p(seen|rel)$ grows with $p(tell|seen, rel)$ (i.e. talking about relevant items implies relevant items are discovered). Similarly, given a fixed x value, the lines corresponding with larger values of $p(tell|seen)$ – and therefore with more negative communication $p(tell|seen, \neg rel)$, since the positive $p(tell|seen, rel)$ is fixed – are in general below the others. Except when the prior $p(tell|seen)$ is quite low (curves in clear blue are below some dark ones when x is close to 1), in those situations is preferable talking about non-relevant items to improve communication and allow these items arise to those users who consider them relevant.

In any case, and despite some small distortions due to low communication levels, there is a clear correspondence between biases. As we just noted earlier, this is not trivial. The explanation is that, intuitively, in a situation with a relevance-prone communication bias, items that many users like find more paths to travel along the network, and therefore reach (be discovered by) more users than items with a lower relevance prior. This intuition also explains the minimum communication level requirement, since this level indicates the average speed at which item information spreads across users. If this is low, the information is not traveling and discovery lies in random exogenous discovery, which of course does not depend on relevance.

The dependency between discovery and relevance was one of the main factors that may affect the recommendations' performance, but not the only one. Another important aspect that comes into play in scenarios with mixed dependency – as the ones produced by this simulation – is the steepness of the discovery distribution. Remind from Chapter 4, that if $p(seen|i)$ is steeper enough than $p(rel|i)$, the effectiveness of (total and relevant) popularity and average rating might move away from optimal values. Moreover, we also proved that strong discovery biases may increase the possibilities of contradiction between observed and true metric values.

Looking to know how the steepness of the discovery distribution varies depending on the communication biases, and in order to later relate such steepness with the recommendation algorithms' performance, Figure 5.3 represents it as the variance of $p(seen|i)$ distribution over the items. In the color map on the left, we show such variance as function of the parameters $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$, in a similar way as in the colour map of Figure 5.2. This time, however, the lowest value (i.e. 0) is coloured in white, and the highest in blue.

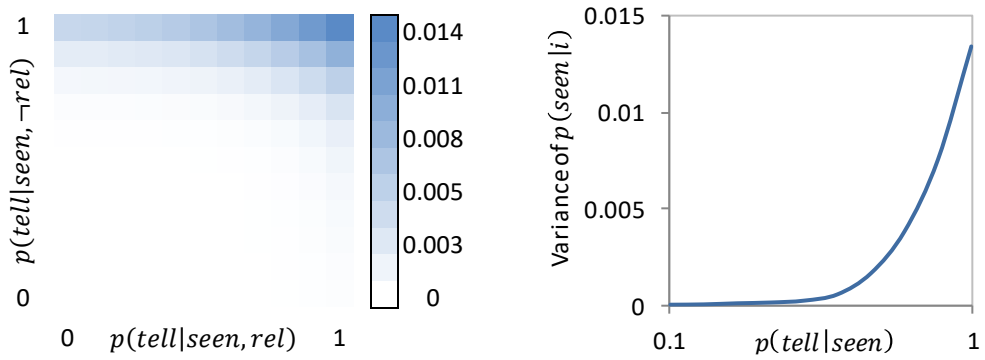

Figure 5.3. Variance of the discovery distribution ($p(seen|i)$ distribution) as function of the communication bias parameters (left) and the communication prior (right).

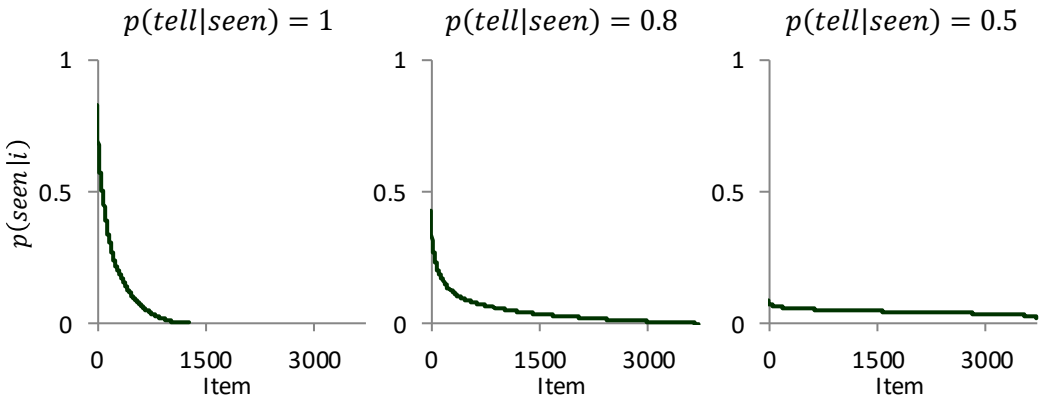

Figure 5.4. Discovery distributions $p(seen|i)$ resulting from situations where users share 100% (left), 80% (middle) and 50% (right) of what they discover.

We can see that in most part of the map the variance is quite low (it is indeed practically 0) and only in the right top corner the colour changes to blue. Therefore, the blue parts (high variance in the resulting discovery distribution) coincide with situations where both sharing parameters present high values. This may suggest that the variance of $p(seen|i)$ distribution increases with the communication level. Note also that a greater value in the negative communication $p(tell|seen, \neg rel)$ seems more important in order to increase variance than the value of relevant communication $p(tell|seen, \neg rel)$. We can check this by noting that the highest row of the map (corresponding with $p(tell|seen, \neg rel) = 1$) is quite darker than the rightest column ($p(tell|seen, rel) = 1$). This is explained by the fact that relevance prior is 28.48% (less than 50%). There are thus more non-relevant tastes than relevant ones, and therefore increasing non-relevant communication raise more the total sharing level than modifying positive communication.

We confirm such correlation between the variance of $p(seen|i)$ and the sharing level – defined by the prior $p(tell|seen)$ – in the right graph of Figure 5.3, where we show the

former as function of the latter. We clearly verify with this graph how the variance is quite low for priors below 0.5, and then it sharply rises when we increase the communication level. To further illustrate this behaviour, Figure 5.4 displays the shape of the resulting discovery distribution in three points from the previous curve, i.e. for communication priors of 1, 0.8 and 0.5. We can see that, for extreme diffusion levels, when users talk about everything they discover (left graph of Figure 5.4), the discovery distribution is quite skewed and more than half of the items have not even been discovered by any user. On the contrary, when users only share 50% of what they discover (right graph of Figure 5.4), we can see that the discovery distribution is essentially flat and that all items are roughly discovered by the same proportion of users.

This behaviour is explained by the same number of discovered (rated) items that determines the final of all simulations. Thus, a higher communication level implies that such number is arisen before, and there is less time for all items to be discovered. Those items sampled first by exogenous discovery will start their propagation before, having more time to being discovered by users than those items sampled later. Moreover, if the communication level is high, the former will spread quickly along the network, accumulating most of the discoveries (i.e. increasing its $p(seen|i)$) and bringing the end of the simulation closer, so when the latter are discovered by the first time there is no time to propagate and they will present a low $p(seen|i)$ (even 0 if they are not sampled at all).

This situation is not artificial, it represents real scenarios where new items are constantly appearing (new films are created, new books are published, new songs are recorded, etc.). As the consequence of these different starting points, items may present quite different discovery ratios and this may have an impact in the recommendations' performance, as we see next.

5.3.2 Effect on recommendation algorithms

We now check the effect of communication biases on recommendations' precision. We do so by the same parameter settings as we just did in previous section, namely, we assume a neutral rating bias and collapse discovery and rating by taking $p(rated|seen, rel) = p(rated|seen, \neg rel) = 1$.

In all the colour maps we display in this section, white colour represents values close to 0, blue indicates positive values and red refers to negative ones. All of them present the relative effectiveness of different recommendation algorithms (mainly compared with random) as functions of the parameters $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$. Note that all the color maps referring observed precision share the same colour scale, so we can know if one algorithm is above other just by observing which one presents the darker blue colour. And the same applies for true precision. Regarding the curves of the right side of the figures, they have the same meaning as the ones shown in Figure 5.2.

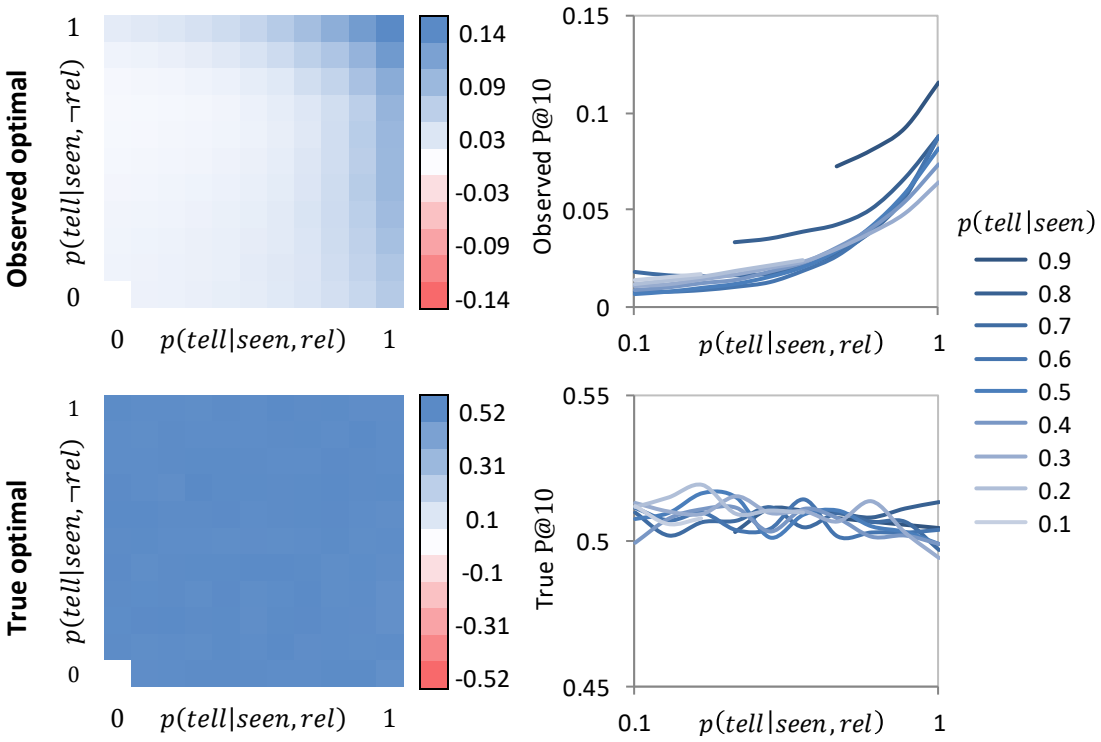

Figure 5.5. Difference between (observed and true) optimal recommenders and random recommendation, in terms of (observed and true, respectively) *P*@10.

6.3.2.1. Optimal rankings

Let us start by analysing the behaviour of the optimal rankings in order to state the superior bounds for the precision of the popularity-based recommendations. Thus, Figure 5.5 shows the difference between optimal rankings and random recommendation. Each optimal is compared in terms of the metric that it optimizes, namely, the difference between observed optimal and random recommendation (top row of Figure 5.5) is computed in terms of observed precision $P@10$, whereas for the true optimal (bottom row) we use true precision $P@10$. Random recommendation works here as a common reference point that allows us to compare several algorithms at the same time that we study their dependence with the configuration parameters.

Before studying the behaviour of each optimal ranking, one first observation that arises when comparing the curves of maximum observed precision (top right graph of Figure 5.5) with the ones of true precision (bottom right), is that the latter shows much larger values than the former. This is because observed precision only counts observed relevance in the form of ratings, which is a fraction of the total relevance that true precision takes into account.

Observed precision

Let us start by commenting the behaviour of the observed ranking (top row). In Chapter 4 (Section 4.7.3), we concluded that the maximum observed precision that can be obtained by a non-personalized recommendation algorithm (i.e. the precision of the observed optimal ranking) increases with the bias of the discovery distribution $p(seen|i)$ (also with the bias of $p(seen, rel|i)$ but let us start with the discovery distribution bias first). This agrees with the behaviour we observe in the top-left colour map of Figure 5.5, since it presents quite the same colour patterns than the one of Figure 5.3, which corresponds to the steepness (bias) of $p(seen|i)$. Note that this explains why, for a fixed value of positive communication $p(tell|seen, rel)$ sharing negative experiences seems to improve the maximum observed precision (see that the columns of the top-left colour map of Figure 5.5 show a darkening down to top). This is because increasing the negative sharing also increases the communication level and, thus, the bias of the discovery distribution.

Another trend we observe regarding maximum observed precision is that it increases when users are prone to share items they like: all the rows of the top-left colour map of Figure 5.5 display a monotonic growth left to right, and all the curves in the top-right graph also show a steady growing trend. This happens because, as we comment before, observed precision also gets higher with the bias of the relevant discovery distribution $p(seen, rel|i)$, which, for a given number of total findings (our simulation stopping condition), increases if discovery is biased towards liked items.

True precision

Regarding the true precision of true optimal ranking (bottom row of Figure 5.5), it does not seem to depend on neither the communication bias parameters (all the colour map is in the same blue tone) nor the sharing level (the curves are all at the same level and draw practically a horizontal line around 0.5). We indeed zoom in on the curves to confirm that they do not present any clear pattern. We explain this behaviour by the strong dependence between the precision of the true optimal ranking and the relevance probability distribution $p(rel|i)$. Remind from Chapter 4 (Section 4.7.3) that this dependency is manifested by the term $p(rel|i)/\big(1 - p(rel|i)\big)$ in the true optimal ranking function, namely, it is a more than linear dependency. Moreover, in our simulation framework, the variance of the relevance distribution is 0.018 (see such distribution in Figure 5.1), a larger value than the greatest variance of $p(seen|i)$, which is ~ 0.014. In other words, the changes in discovery distribution (caused by different communication patterns) are not big enough to alter the behaviour of the optimal true ranking, which is thus governed by the same relevance distribution in all the simulations.

This makes us think about whether this simulated behaviour represents a real situation or not, namely, whether discovery distribution is typically much less steep than relevance distribution or not. In fact, it is. In real scenarios, there is typically a huge amount

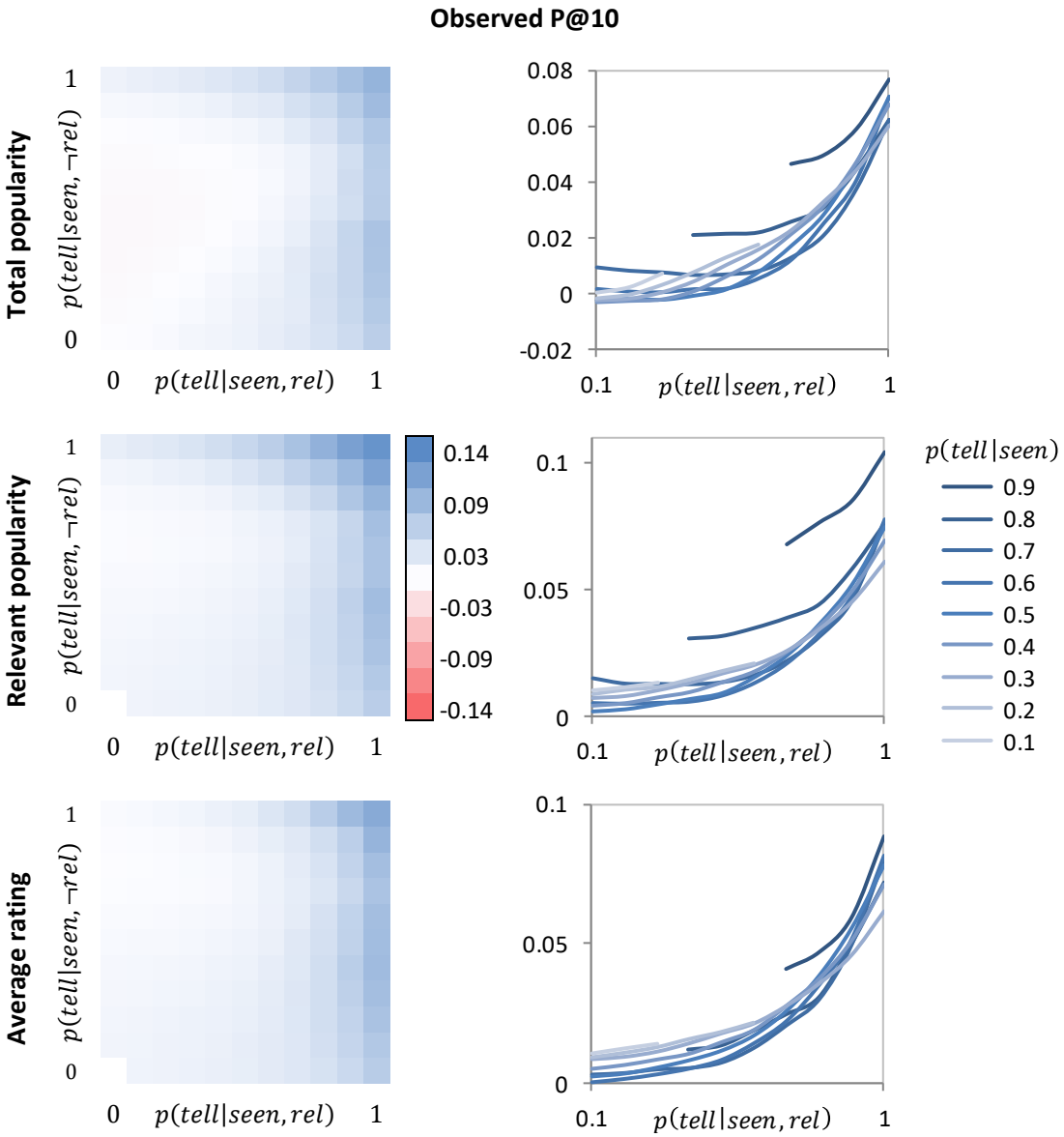

Figure 5.6. Difference between popularity-based recommenders and random recommendation, in terms of observed *P*@10.

of options that users can choose. However, due to a variety of reasons – mainly time but also others as the inability to process so much information – they are only able to discover a small fraction of these options, giving rise to a discovery distribution of small dimensions versus the ones of the ideal full relevance distribution. It is thus expected that maximum true precision is governed mainly by this user's taste distribution.

6.3.2.2. Popularity-based recommendations

Once we have stated the maximum (observed and true) precision values, as well as studied how they depend (or not) on the communication biases, let us analyse now the effect of such biases in the performance of the popularity-based variants: total popularity, relevant

popularity and average rating. The latter, average rating, is smoothed using an additive smoothing in the same way as we did in the experiments of Chapter 4 (Section 4.8.2). Figure 5.6 shows the difference in observed precision between these three algorithms and random recommendation, whereas the comparison in terms of true precision is depicted in Figure 5.7. The first, second and third row of both figures refers to the relative effectiveness of total popularity, relevant popularity and average rating, respectively.

Observed precision

Let us focus first on the observed precision of these recommendation algorithms, i.e. in Figure 5.6. We see that they present quite a similar behaviour than maximum observed precision (see top row of Figure 5.5), namely, they grow with the communication level in general, and with the relevance-prone telling bias once that such level is fixed. This similar behaviour is especially remarkable in the case of relevant popularity. Moreover, its absolute precision values are very close to the maximum ones (we note this when comparing the curves for different sharing priors, using the right graphs of both figures, and also in the dark blue tone of the corresponding colour maps). The patterns of total popularity are also quite similar to the ones of the optimal observed ranking, however, it does not reach so high precision levels – although not for a great difference – and it is even worse than random (red colour in the colour map and a negative value in the curves) when the communication is low and biased towards non-relevant items.

Average rating presents some more differences. For instance, it seems that the communication level does not favour so much this recommendation algorithm as popularities (note that in the right curve graph of the third row of Figure 5.6 the curves corresponding to the sharing priors 0.9 and 0.8 are not so high with respect to the other priors as the ones of total and relevant popularity in the first and second rows of Figure 5.6, respectively). This is probably because average rating does not depend so much on discovery distribution as on relevance distribution (as it happened with true optimal ranking), and the latter is the same in all simulations. We will compare in more detail average rating and relevant popularity later.

True precision

The differences between these three popularity variants arise when we compare them in terms of true precision (see Figure 5.7). Total popularity (first row) results to be once again quite an unreliable option. We indeed corroborate here our conclusions of previous chapter, where we stated that the effectiveness of this recommend mainly depends on the sign of the comparison $p(seen|rel) > p(seen|\neg rel)$, namely, on whether relevant items are more discovered than non-relevant ones or less. As we showed in the previous Section 5.3.1, discovery biases are mainly determined by the communication ones, and then the sign of the previous comparison is equivalent to the sign of the following one: $p(tell|seen, rel) > p(tell|seen, \neg rel)$. Thus, we observe that red color dominates in

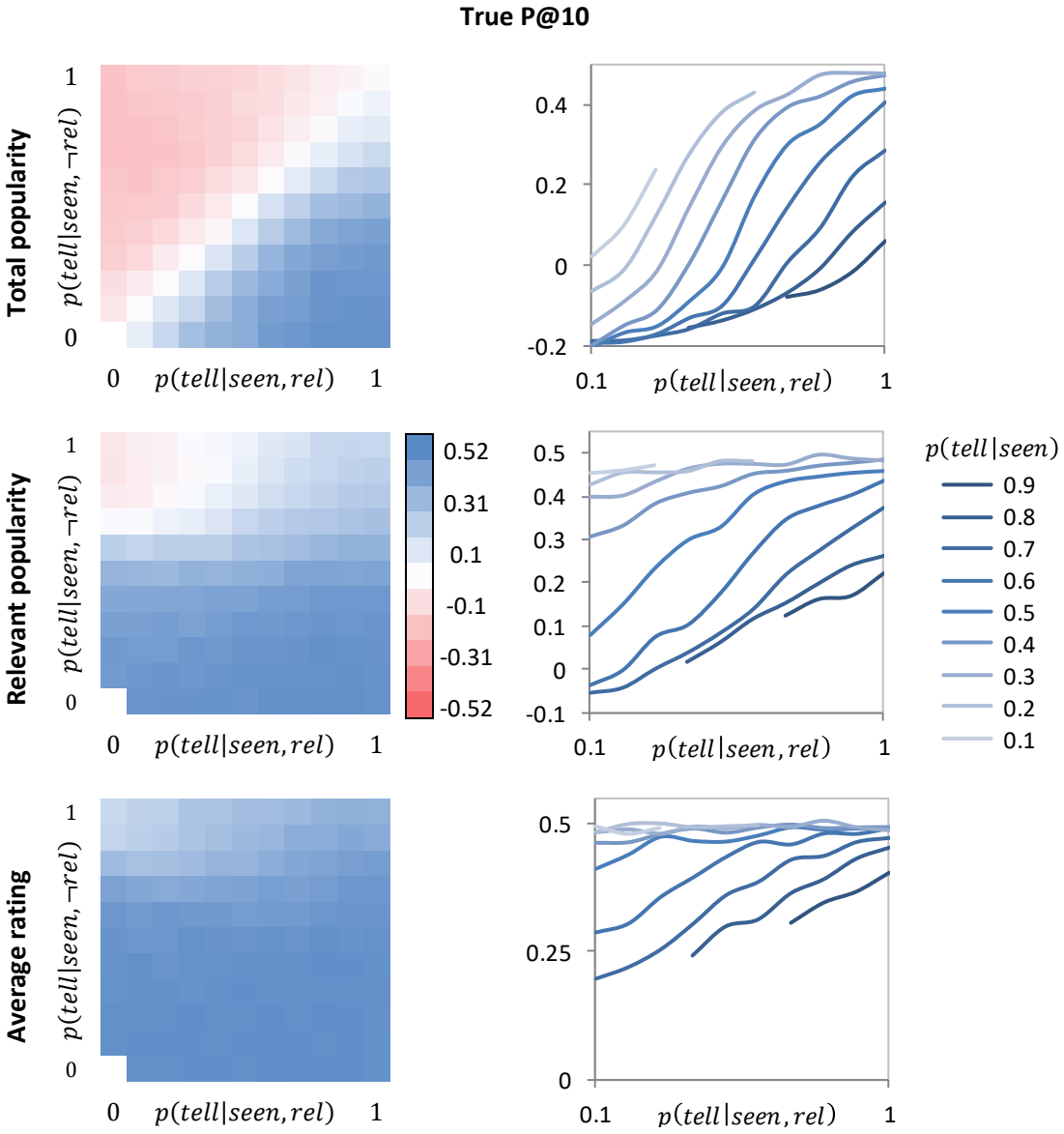

Figure 5.7. Difference between popularity-based recommenders and random recommendation, in terms of true *P*@10.

the left top triangular matrix (corresponding with $p(tell|seen, rel) < p(tell|seen, \neg rel)$) while the right bottom zone ($p(tell|seen, rel) > p(tell|seen, \neg rel)$) is mainly blue. Remind that red indicates negative values and blue indicate positive ones, so precision of total popularity is below or above random on each side of the diagonal. In fact, the darkest colours (both red and blue) are reached in the top left and bottom right corners, when the difference between $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$ is bigger.

This trend of total popularity is explained because in a relevance-biased communication, the number of total ratings of each item will correlate with the number of users who

like each item. Thereby liked items become statistically more popular, causing an increase in the resulting true precision of total popularity recommendation. In the opposite case, the items with most ratings (predominantly negative) are not liked by many users, yet they get recommended by total popularity.

Relevant popularity on the other hand is more robust against this trend: it is almost insensitive to the relevance bias for $p(tell|seen) < 0.5$, since the precision curves for such low levels of global communication (right graph of the second row of Figure 5.7) run almost constant and high with respect to $p(tell|seen, rel)$. They are indeed quite close to the maximum true values, as we can note by comparing these curves with the ones in the second row of Figure 5.5. This makes sense, since the lower the communication levels, the lower the bias of discovery distribution, and therefore the items are discovered (and thus rated) to similar extent. In this situation of low discovery bias, relevant popularity can correctly identify the relevant items by correlation with positive ratings. Even if $p(tell|seen, rel) < p(tell|seen, \neg rel)$, true precision is still good because negative ratings are ignored by this variant, for which the only thing that matters is the correlation between relevance and relevant rating.

However, when the communication level grows, the discovery becomes biased, and in this situations sharing non-relevant items above a certain degree can harm relevant popularity: the curves corresponding with high sharing priors ($p(tell|seen) > 0.5$) indicate a poor performance (even worse than random) when $p(tell|seen, rel)$ is low. This happens because, due to the sharing bias against relevance, those items which are not liked by many users are the most discovered ones. They can thus get enough positive ratings to surpass other more relevant items, for which relevance remains more unobserved due to the great discovery bias.

Regarding average rating (last row of Figure 5.7), it is the only popularity-based algorithm that remains always above random recommendation (in terms of true precision), no matters how strong the communication biases are. However, observing the patterns of the colour map and the curves, we can note that it is somehow affected by similar trends as the ones of relevant popularity. Namely, with high sharing priors and a communication bias against relevance, its effectiveness decreases, moving away from the maximum true precision values. This decrease is nevertheless smaller than in relevant popularity, and indeed the precision values of average rating seem to be consistently higher than the ones of relevant popularity, closer to the maximum ones.

6.3.2.3. Relevant popularity vs. average rating

In order to verify this last observation about the differences between average rating and relevant popularity, in Figure 5.8 we explicitly report such differences in terms of observed (top) and true (bottom) precision. We use again a colour map and a curve graph that follow the same structure and meaning as in the previous figures. The only difference is

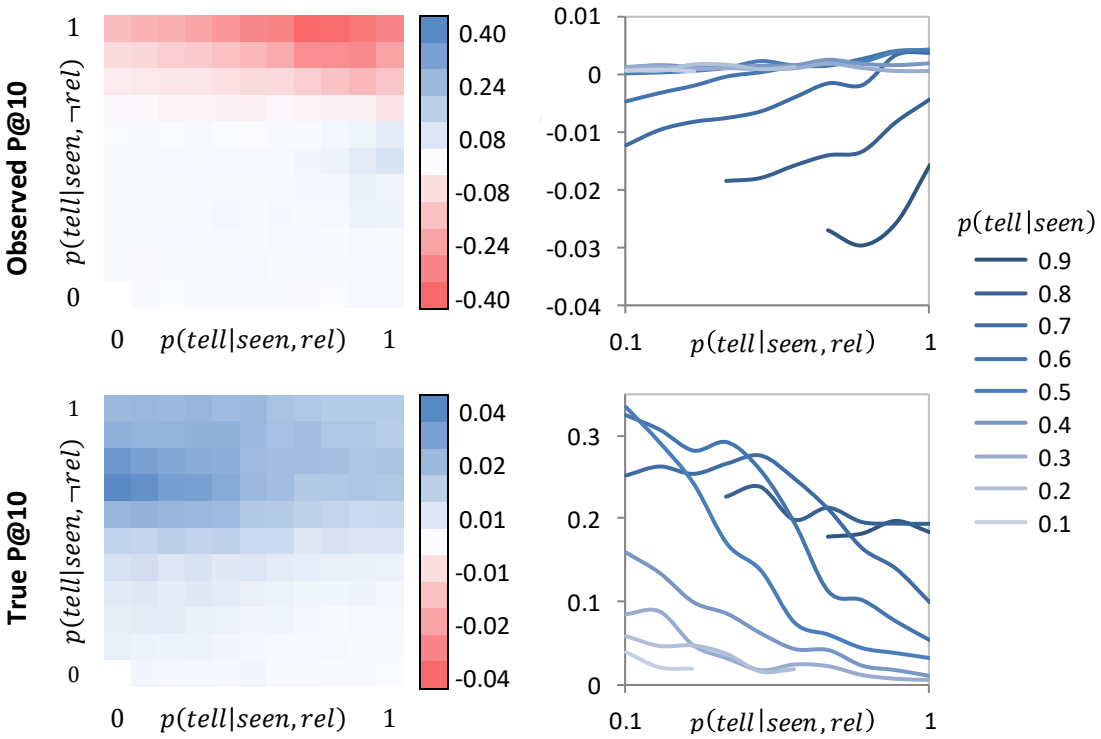

Figure 5.8. Difference between average rating and relevant popularity, in terms of true and observed *P*@10.

that, this time, the values depicted are the average rating precision less the precision of relevant popularity.

Looking at the bottom row, we can corroborate that all displayed values are positive. Hence, in terms of true precision, average rating is always the best option. Its advantage is indeed larger when users mainly talk about what they dislike (columns in the colour map show a darkening from down to top), probably because this behaviour pattern is particularly harmful with relevant popularity.

However, that situation is not what we will measure if we only have observed precision to compare, as is the case of most offline experiments. If we take a look to the top graphs of Figure 5.8, we note that observed precision is reporting quite the opposite message than true precision: average rating is indeed marked as worse right where it is really much better. In particular, high communication (and thus discovery) levels seem to unfairly reward relevant popularity, increasing its apparent advantage to average rating (the curves taking negative values are the darkest ones, corresponding with highest sharing priors). This confirms once again the theoretical results of Chapter 4, where we already proved that discovery biases promote the emergence of contradictions between observed and true metrics, mainly because they cause an unfairly reward in the observed performance of relevant popularity.

Before moving to analyse the rating bias effects, let us try to summarize the main observations of this section. First, observed precision is mainly dominated by sharing (discovery) biases, and more specifically by the communication level. In fact, the observed precision values of all recommendation algorithms, but especially of popularities, increases with the bias of discovery distribution, as well as with relevance-prone sharing trends.

On the contrary, maximum true precision is not altered by any sharing decision pattern. However, high discovery levels, combined with biases against relevance, might distort the real popularity-based recommendations' performance making some of them obtaining worse results than even random. Average rating is the recommendation algorithm which presents most robustness against this kind of trends. It is indeed above the other two options in all the situations, in terms of true precision. Regarding the comparison total vs. relevant popularity, the latter is clearly a much more reliable option, both in terms of observed and true precision.

Finally, an important highlight that arises from the previous two paragraphs is once again the existing contradiction between observed and true metrics. In the simulated experiments of this section, they present the opposite behaviour with regard to the communication level (i.e. discovery bias distribution): observed precision wrongly rewards recommendations when this level is high, whereas what is really happening is that they are performing worse. Moreover, the observed reward is mainly focused on relevant popularity, making it appears better than average rating, when it is the latter which is indeed being more robust to the effect of large discovery biases.

5.4 Rating bias

We now move to analyse how user behaviour biases in rating decision might affect the algorithms' performance. As we did before, we may pretend to study in isolating these effects, by neutralizing any potential discovery bias. However, a perfect neutralization is not possible with our simulation framework. We can remove relevance dependency by assigning $p(tell|seen, rel) = p(tell|seen, \neg rel)$, but the item dependency still remains as far as different items are discovered by the first time at different moments, and therefore some items have more time to be discovered than others. This gives rise to a biased discovery distribution, whose bias is determined by the communication level, as we verify in previous Section 5.3.1. The lower the sharing prior, the smaller the discovery distribution bias, and thus, the closer to (but without reaching) a situation of neutral discovery. Note that this is not an artificial behaviour, in real scenarios is practically impossible to observe a complete flat discovery distribution (i.e. all items being discovery to the same extent).

Therefore, we shall be careful with simply assigning both previous probabilities $p(tell|seen, rel)$ and $p(tell|seen, \neg rel)$ to 1, since then the high communication level will bias the discovery distribution and might significantly alter the results. For this reason, apart from the previous extreme diffusion setting we will also consider a scenario with a flatter discovery distribution by assigning a lower communication rate of 0.5.

Extreme diffusion level

Figure 5.9 shows the recommendations' effectiveness in the case of extreme diffusion level, in terms of observed and true precision. All values represent the difference with the effectiveness of random recommendation. In the graphs, we vary $p(rated|seen, rel)$ and analyze the resulting curves for fixed values of $p(rated|seen)$. This time we remove the corresponding colour maps since they do not contribute with any finding which cannot be appreciated with the curves.

The first obvious trend is that observed precision (left colum of Figure 5.9) of all recommendation algorithms, including the optimal one, grows with the bias towards rating relevant items. This is because, for a fixed total number of generated ratings (the simulation stopping condition), the bias of the relevant rating distribution $p(rated, rel|i)$ increases with $p(rated|seen, rel)$. And as pointed out earlier, the bias of $p(rated, rel|i)$ increases the maximum observed precision. To be precise, we proved that it increases with the bias of the relevant discovery distribution $p(seen, rel|i)$, but an analogous result aplied to $p(rated, rel|i)$ can be derived if we develop the demostrations of Chapter 4 (Section 4.7.3) before introducing discovery distribution.

This effect is not observed in true precision (right colum of Figure 5.9). As in the discovery bias study, maximum true precision does not depend on any rating decision bias, since it is mainly dominated by the releance distribution (which is the same in all simulations). However, and quite paradoxically, the true precision of popularities and average rating degrades with the positive rating bias (the curves display a decreasing trend). This is explained by a non-trivial interaction between a viral network effect and the exclusion of already rated items, as we explain next.

The communication setting in these experiments is of extreme diffusion, giving rise to a strong biased discovery (and thus rating) distribution, as we verify in Section 5.3.1. In other words, some few items will accumulate a comparatively high number of ratings. Moreover, due to item exclusion, these most rated items will be excluded from the rankings offered to all the users who have rated them. As $p(rated|seen, rel)$ grows, only users liking these top viral items will rate them, excluding such items from their rankings. The more relevant the item, the more rated it is, but only by users who like it. The number of relevant ratings of these items will thus increase, so any recommendation algorithm that considers such number as usefull signal will try to recommend them, and only will be able to do it for those users who actualy do not like them. Consequently, the true precision of this algorithm will decrease. Note that this is a special situation where

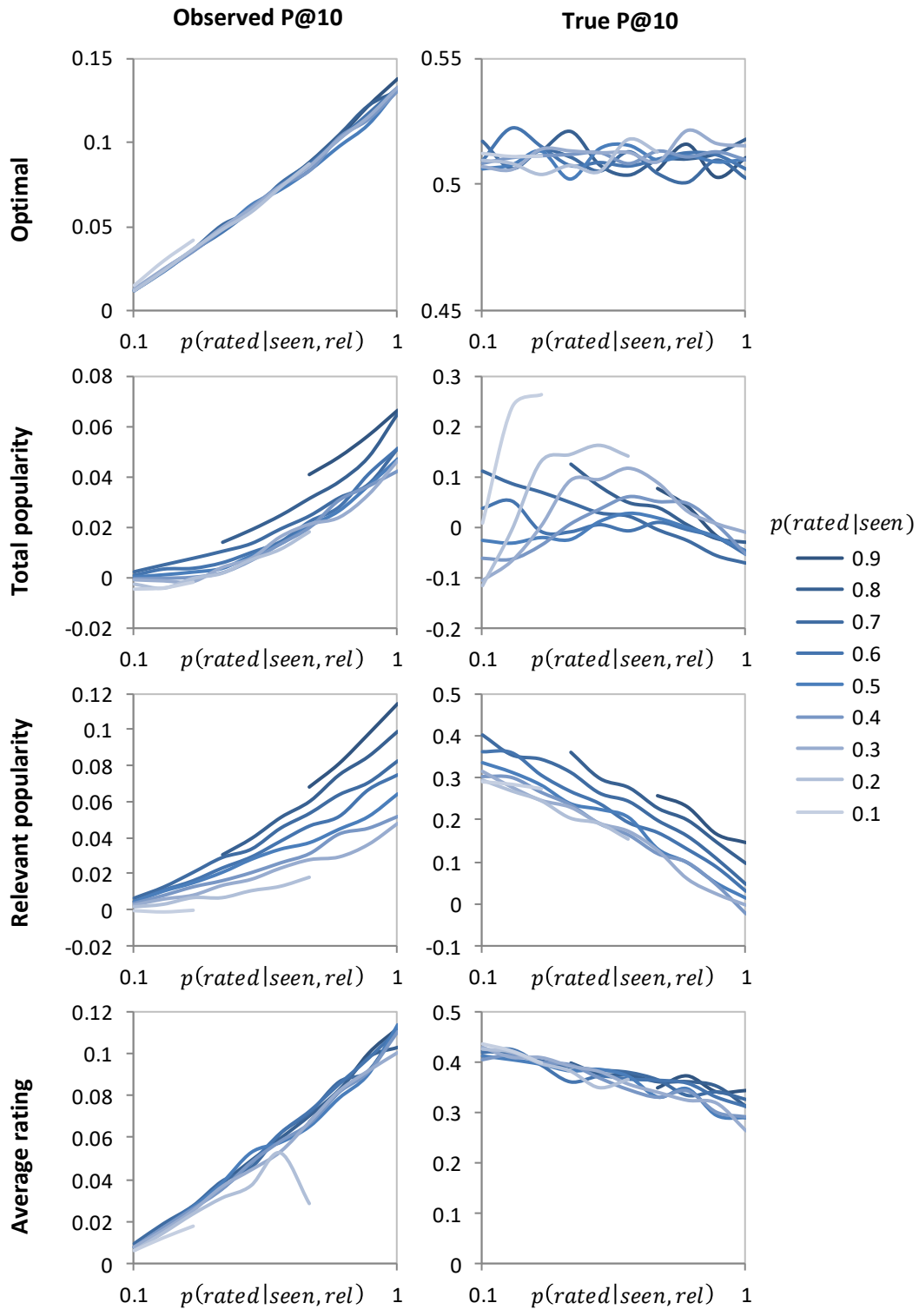

Figure 5.9. Difference of optimal and popularity-based recommenders with random recommendation, in terms of observed and true $P@10$, and under a scenario of extreme diffusion, i.e. $p(tell|seen, rel) = p(tell|seen, \neg rel) = 1$.

recommending acording to the underlying total relevance distribution $p(rel|i)$ is not optimal and, due to item exclusion, items that are unknown mainly by users who like them are better candidates than others with a higher relevance probability.

Unfortunately, most recommendation algorithms use the number of relevant ratings to some extent, in particular the three popularity-based recommendations that we are considering (total popularity also does it because the number of total ratings usually presents certain positive correlation with the number of relevant ratings) . Average rating is once again the algorithm which seems more robust against this phenomenon, by manteining a large distance from random.

Note also that, in the case of relevant popularity, an increase in the rating level produces and notable improvement on its true precision: darker curves are clearly above the rest. The reason is that, for a given $p(rated|seen, rel)$ value, if we increase the rating prior, we are increasing the number of non-relevant ratings per item. Relevant popularity ignores this value and keeps recommending the same items (those with most relevant ratings), but the extra negative ratings avoid failures to this method. Thus, this time the most popular items are not going to be recommeded to those new users who have rated them with a negative value, and that otherwise would have produced a decrease in precision. In the case of average rating and total popularity this effect is not observed (dark curves are mixed with clear ones), since they do indeed consider the number of negative ratings to some extent and the recommendations may change when increasing such number.

We already predicted the effects descrived in previous paragraphs in Chapter 4 (Section 4.7.3) when stating that, in general, items with a high $p(seen|\neg rel, i)$ and a low $p(seen|rel, i)$ are preferable in terms of true precision. If we substitute $seen$ by $rated$ the statemente is also true, and means that it is better to recommend items rated by users that do not like them (since this avoids failures) and unknown by users who do like them (since this causes successes).

Moderate diffusion level

Now, let us see what happen when we consider a lower communication level. Figure 5.10 shows the recommendations' observed and true performance in such case. We see that the paradoxical effect of the positive rating bias in the true precision of popularity-based variants disappears (right column of Figure 5.10). Now, in fact, there is no dependence on the bias in relevant popularity and average rating, making them closer to maximum true values. The reason is that, in this experiment, the diffusion is not so strong, giving time to all items to be discovered (and thus rated), and producing a less biased discovery and rating distributions. The more even rating distribution reduces in turn the effects of item exclusion described before, since now most relevant items are not so broadly excluded, and lets true precision being governed by the full relevance distribution. In other words, recommending most relevant items is this time a good option.

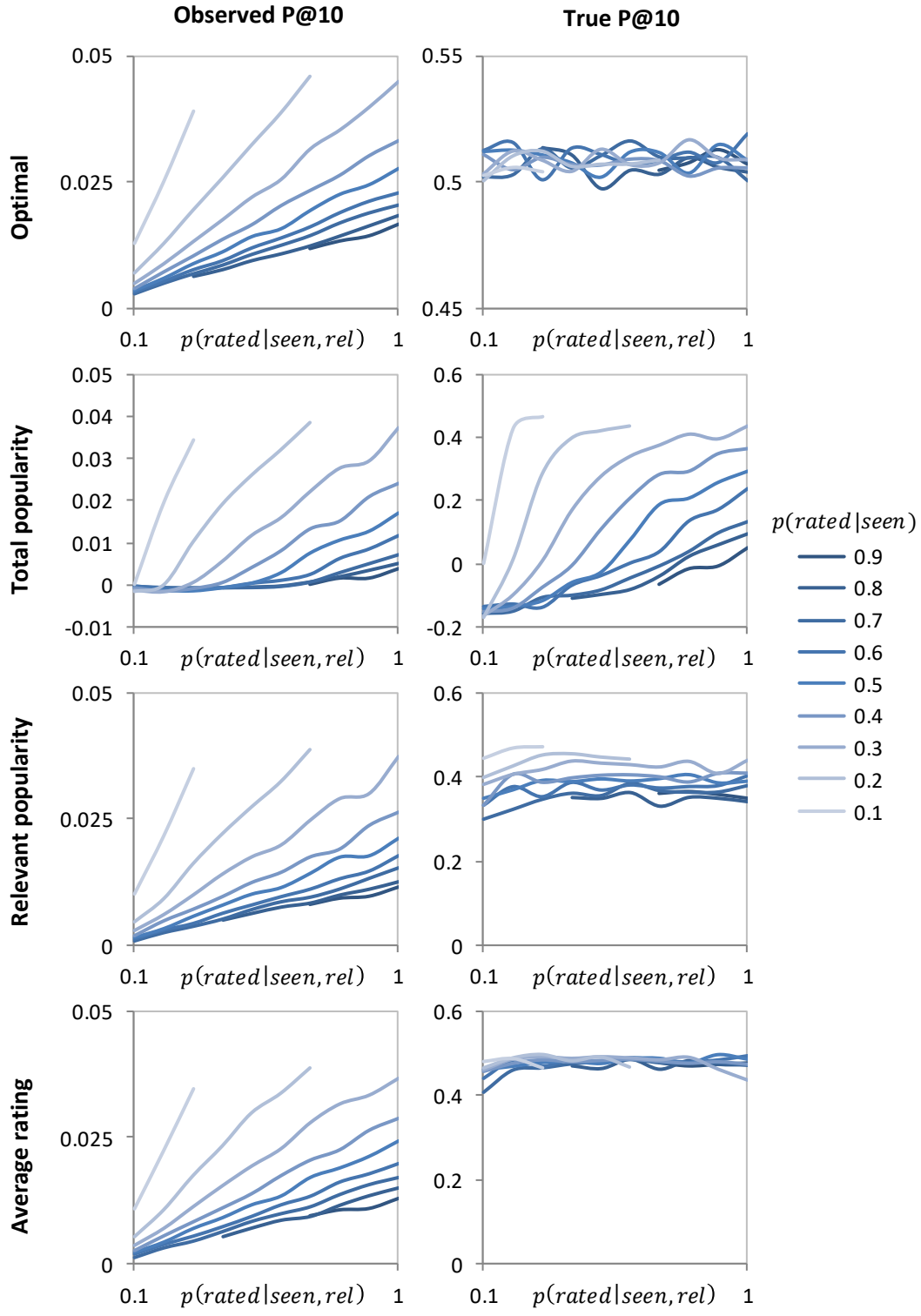

Figure 5.10. Difference of optimal and popularity-based recommenders with random recommendation, in terms of observed and true $P@10$, and under a moderate communication level: $p(tell|seen, rel) = p(tell|seen, \neg rel) = 0.5$.

Despite the correlation between relevant rating and relevance (the most relevant the item, the most users will rate it with a positive rating), the average rating seems to identify the relevant items slightly better than does relevant popularity, probably due to its stronger dependency on the relevance distribution. Regarding the true precision of total popularity, it does depend on the ratio of positive vs. negative ratings, and this is clearly shown in the corresponding graphic, where in fact precision steps up as soon as $p(rated|seen, rel) > p(rated|seen, \neg rel)$.

Finally, the observed precision of recommendation algorithms (left colum of Figure 5.10) increases with $p(rated|seen, rel)$, as in the extreme communication case, because it raises the bias of $p(rated, rel|i)$ distribution.

5.5 Networks effects

As we have seen in the previous section, in addition to the effect of individual users' behaviour, further network effects as the diffusion speed may emerge from social-level dynamics, which end up affecting algorithm's performance.

Now, we examine whether the network structure can be another factor that alters the observed dynamics and thus the recommendation algorithms' behaviour. For this purpose, we run the same simulation on a Barabási-Albert graph (Barabási & Albert 1999) with the same number of nodes (users) and edges (friendship links) as in the Facebook dataset.

The network structure only intervenes in the sharing decision process, constraining the friends with which a user can talk about an item. Consequently, it might directly alter the shape of the discovery distribution, but any influence it could have on the rating distribution is due to the dependence between rating and discovery (users can only rate what they know). For this reason, we collapse discovery and rating by assigning $p(rated|seen, rel) = (rated|seen, \neg rel) = 1$. Moreover, we take a configuration of extreme communication level (i.e. $p(tell|seen, rel) = p(tell|seen, \neg rel) = 1$). In other words, users share and rate everything they discover. The reason of this last assignment is that, intuitively, the constraint in the maximum number of contacts per user (i.e. the influence of the network structure) must be more noticeable on those situations where users try to talk more often with their friends.

Thus, Figure 5.11 shows the positive rating, discovery and relevance distributions obtained when running the simulation with the previous parameter configuration and using as social network structure the Facebook data (left) and the Barabási-Albert graph of equivalent proportions (right). The $\boldsymbol{x}$ axis of both graphs corresponds to the items, sorted by the number of users who have discovered them. Each dot shows the ratio of users who like (red), have discovered (green), and have rated positively (yellow) the corresponding item.

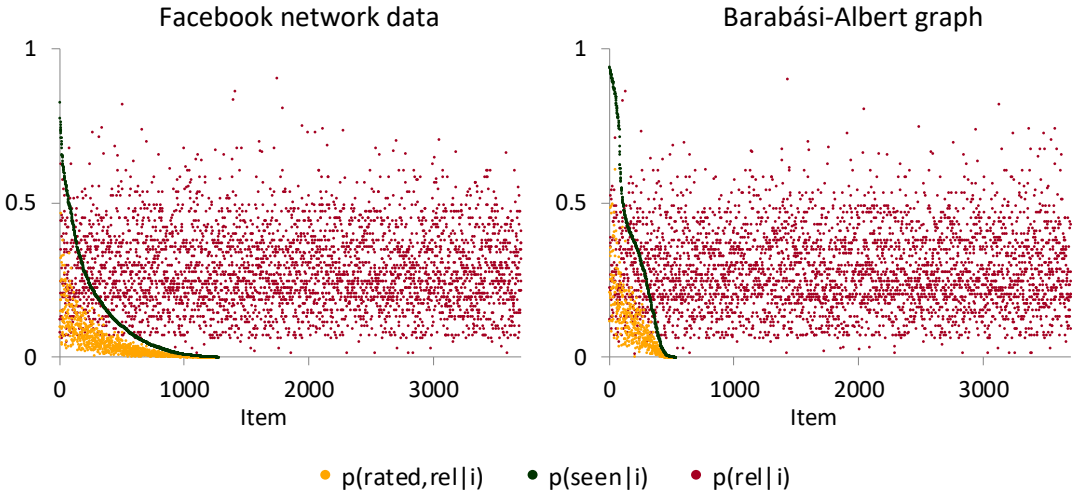

Figure 5.11. Positive rating, discovery and relevance distributions – i.e. $p(rated, rel|i)$, $p(seen|i)$ and $p(rel|i)$ respectively – obtained with different social network structures: Facebook data (left) and a Barabási-Albert graph of equivalent dimensions (right).

The most remarkable aspect to point out is that discovery distribution is steeper when using the artificial Barabási-Albert graph (right) than with the real one of Facebook (left). In the case of the Barabási-Albert network, the best-known items have been discovered by almost all users, and indeed only around 500 items are known by some user. The rest have not been discovered at all. The distribution is also biased in the Facebook graph, but even the most known items still present a margin to new potential findings, and more than 1000 items have been discovered to some extent. This different bias in the discovery distribution implies that the information travels faster on the preferential attachment model of Barabási-Albert. In addition, the shape of the green distribution, with two elevations that are not present in the corresponding curve of the Facebook graph, seems to suggest the existence of two quite differentiated clusters of users.

Note also that discovery is neutral with respect to relevance in this configuration (hence the green and red plots do not show correlation). The positive rating distribution correlates with discovery because the more an item is discovered (and thus rated, under the collapse between discovery and rating) the more chances it gets to obtain a relevant rating.

Figure 5.12 shows the effect of this in the resulting precision. The results in terms of true precision do not differ significantly between the two types of graphs, except for the inversion between total popularity and random, but both algorithms are quite proximal in the two cases and it is most likely due to chance. Also, the true difference between relevant popularity and average rating slightly increases when changing to Barabási-Albert graph, probably because the viral effect is stronger in this graph, and average rating is more robust against it, as we have seen in previous section.

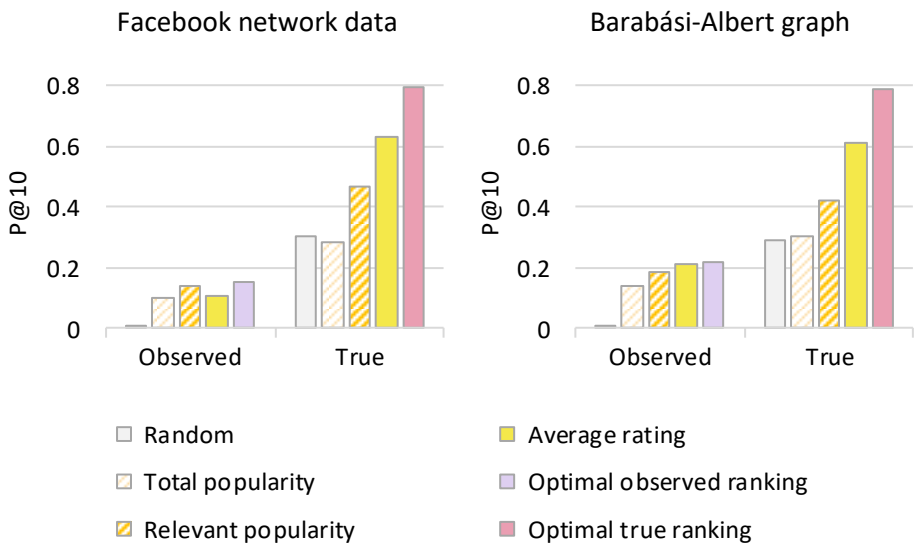

Figure 5.12. Recommenders' performance obtained using as social network a Facebook data (left) and a Barabási-Albert graph of equivalent dimensions (right).

Notwithstanding, the most relevant result of Figure 5.12 refers to observed precision. Thus, we can see that average rating is below relevant popularity when using Facebook social network structure, but above it when we change to the Barabási-Albert model. This implies that the contradiction between observed and true precision that exists on Facebook (observed precision is telling that relevant popularity is better when average rating is indeed obtaining the higher true precision), disappears in the other graph. In other words, with the network structure of Facebook (unfortunately the one that comes from a real social network), in an offline evaluation experiment we would have wrongly concluded that relevant popularity is the algorithm which performs the best, whereas using the Barabási-Albert graph we would have got it right and determined that average rating is better.

We conclude therefore that the social network structure can be decisive when evaluating recommender systems in a social environment, it might distort the results to the point of misleading the determination of the best algorithm.

5.6 Conclusions

In this chapter, we have extended the rating generation probabilistic model of Chapter 4 to include the transfer of item information between users. Moreover, we have incorporated to the model a series of dynamics and action sequences that allow us to simulate the generation of ratings in a social network scope.

By such simulation, we have been able to recreate several scenarios of mixed dependencies between discovery, relevance and items, and study the behaviour of recommender

systems on these situations. The main conclusions arisen from this study are summarized below:

- Observed precision, for both optimal and popularity-based recommendations, grows with behaviours that favour a biased relevant rating distribution. Thus, it increases with the discovery bias (i.e. high communication priors) as well as with the relevance-prone rating and sharing biases. On the contrary, maximum true precision does not seem to depend on any bias, neither sharing nor rating user behaviours affect their value. It depends mainly on the relevance distribution, and thus is not affected by rating or discovery biases.
- How quickly the information spreads across the social network may significantly affect the true performance of popularity-based recommendations. If the difussion is too fast, a combinacion of dependecies (fom relevance in one hand, and from first discovered items in the other) bias the discovery distribution providing with misleading signals to algorithms, which fails in their recommendations. For instance, viral propagation enhances the effects of item exclusion, so combining it with a bias towards rating liked items results in a counterintuitive decrease of true precision. However, average rating is once again the most robust recommendation algorithm against all these behaviours, confirming the theoretical findings of Chapter 4.
  On the contrary, when the communication bias is moderate, average rating and relevant popularity tend to ignore biases, behaving in a similar way as the optimal ranking.
- Regarding the comparison between the three popularity-based variants, we have verified the general trends that we deduce for mixed dependence scenarios in previous chapter. That is, in terms of true precision, average rating is above relevant popularity in all the studied situations, thus establishing itself as the best popularity-notion despite observed precision is very often telling quite the opposite message. Regarding total popularity, it is confirmed in all the configurations and experiments as a very poor and fragile recommendation algorithm comparing with relevant popularity. Indeed, one of the most obvious conclusions of this chapter and the previous one is that, if we choose to recommend by popularity defined as the number of ratings of each item instead of its average rating, we must focus on relevant ratings.
- Viral propagation boosts observed precision of all recommendation algorithms, but quite especially that of relevant popularity, which quite often results in a disagreement between measured and true precision when comparing this algorithm with average rating. This implies once again that the typical results we are observing in common offline experiments, and that place relevant popularity well above average rating, must be revised and re-evaluated with unbiased relevance samples. Otherwise, the rating data we are using to evaluate might be distorted by discovery biases and producing wrong evaluation outcomes.

- The social network structure may significantly alter the information spread dynamics, and therefore the discovery distribution and the algorithms' behaviour. We have seen, for instance, that real social networks seem to moderate the diffusion level, and thus its consequences on popularity.

Chapter 6

Popularity biases in the nearest neighbours approach

After analysing the potential effectiveness of majority-based rankings as recommendation criteria, and the possible popularity biases in offline evaluation, we now turn to analyse and seek a better understanding of the potential popularity bias in collaborative filtering. We empirically illustrated such biases with examples in Chapter 2. We now seek a formal confirmation and a deeper explanation for the biases.

We do so for a specific and possibly best-known collaborative filtering algorithm: k nearest neighbours (kNN). Our approach is based on a full probabilistic formalization of the kNN scheme, upon which we will evidence the structural presence of (different variants of) popularity – in (different variants of) kNN. A substantial part of the work presented in the current chapter was published in Cañamares and Castells (2017a).

6.1 The nearest neighbour approach

As we already introduced in Chapter 3, the k nearest neighbours (kNN) approach belongs – together with matrix factorization – to the group of collaborative filtering methods. Namely, it employs the manifested opinions of other users, and the similarities that they present with the own user appraisals, to guess her preferences over those items she has not interacted with. Then, these guessed preferences will be used by the algorithm to rank the items and produce the final recommendation.

Using the notation $r(u, i)$ introduced in Chapter 2 to refer to the observed rating by a user u for an item i, let us denote by $\hat{r}(u, i)$ the score function computed by the system to rank the items. Note that $\hat{r}(u, i)$ is not necessarily a rating prediction, it could be any function that aims to guess the preference level of a user over an item, and that therefore may be used to rank the items. We also use the convention $r(u, i) = 0$ to indicate $r(u, i)$ is unknown to the system.

Using previous notation, the most common formalization of the kNN ranking function is given by the following expression:

$$\hat{r}(u,i) = C \sum_{v \in N_k[u]} sim(u,v) r(v,i) \tag{6.1}$$

where C is a normalizing constant whose typical values we comment later; $sim(u,v)$ is a similarity function that quantifies how close the opinions of two users are; and $N_k[u]$ is a subset (neighbourhood) of k users who have been selected based on their suitability as advisors to u. In practice, $N_k[u]$ is formed by the most similar users of u in terms of $sim(u,v)$, but nothing prevents us from selecting such neighbourhood using any other criterion.

Translating Equation 6.1 to words, kNN estimates the preference level of the target user u to a specific item i as a weighted sum of her neighbours' ratings (neighbours means users with some level of similarity with u), where the weight of each rating is the similarity between the corresponding neighbour and the user. Note that the underlying assumption of kNN is that users with similar choices in the past may enjoy similar items in the future, an assumption that we will discuss later.

Another possibility for modelling this assumption is to take an item-oriented approach, and estimate $\hat{r}(u,i)$ as a weighted sum of other ratings of u, weighting each rating value by the similarity between the recipient item of the rating and i:

$$\hat{r}(u,i) = C \sum_{j \in N_k[i]} sim(i,j) r(u,j) \tag{6.2}$$

Regarding the similarity function, there are numerous and different ways to define it, both for users and items. The most commonly used are the cosine function and Pearson correlation. In this chapter we are going to use the cosine similarity as a frame of reference. Such similarity is defined as:

$$sim(u,v) = \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sqrt{\sum_{j \in \mathcal{J}} r(u,j)^2} \sqrt{\sum_{j \in \mathcal{J}} r(v,j)^2}} \tag{6.3}$$

$$sim(i,j) = \frac{\sum_{v \in \mathcal{U}} r(v,i) r(v,j)}{\sqrt{\sum_{v \in \mathcal{U}} r(v,i)^2} \sqrt{\sum_{v \in \mathcal{U}} r(v,j)^2}} \tag{6.4}$$

A final point of variation in the schemes described by Equations 6.1 and 6.2 is the value of the constant c. Most of the literature (including popular surveys as the ones published by Adomavicius & Tuzhilin in 2005 or Ning et al. in 2015) reports taking a normalized value for this constant, in order to convert the sums of Equations 6.1 and 6.2 into weighted averages, namely, make the rating weights add to 1. This normalization makes it possible to consider the resulting score $\hat{r}(u,i)$ as prediction of the rating value

that the user u would have assigned to the item i. Thus, for the user-based approach such normalizing value is given by:

$$C = 1 \Big/ \sum_{v \in N_k[u]:\, r(v,i) \neq \emptyset} |sim(u,v)|$$

while for the item-based variant results in:

$$C = 1 \Big/ \sum_{j \in N_k[i]:\, r(u,j) \neq \emptyset} |sim(i,j)|$$

Another value recently considered for this constant consists on simply taking $C = 1$ for both user-based and item-based versions. We will refer to the first option as normalized kNN and use the term non-normalized for the second one. As we introduced in Chapter 2, numerous recent experiments show that non-normalized variants are generally more effective than normalized ones at the item ranking task (Aiolli 2013, Cremonesi et al. 2010, Vargas & Castells 2014). Indeed, the extended use of the latter come from the recommender systems origins, when the recommendation task was considered as a rating prediction – instead of an item ranking – task.

The main limitation of the previous kNN schemes is their heuristic nature. In fact, heuristic is a name often used to refer to them (Adomavicius & Tuzhilin 2005). Thus, the kNN formulation is supported by the intuition we introduce before (users with similar behaviour in the past should have similar behaviour in the future) and its empirical effectiveness, but it does not arise from a formal justification. Consequently, it is not possible to explain the kNN behaviour in a principled way, neither its effectiveness nor its potential connection with popularity.

To address this issue, in the following section we propose a probabilistic representation of all the previous heuristic variants (user-based, item-based, normalized and non-normalized). A representation that allows us to connect such algorithms with the different popularity variants. Thus, we will find that some variants present a structural bias towards relevant popularity whereas others are guided by average rating.

6.2 A probabilistic formulation of kNN

We start our derivation of a probabilistic kNN variant by modelling the dynamics that govern user-item interactions as random processes in a sampling space. Accordingly, let us consider the (abstract) set Ω of all potential interactions between user and items that might take place at the next unit time. In Figure 6.1, we represent this set Ω as an urn in order to facilitate the explanation. Now let us suppose that we can estimate the probability that each interaction takes place at the next moment, and that we sample one interaction

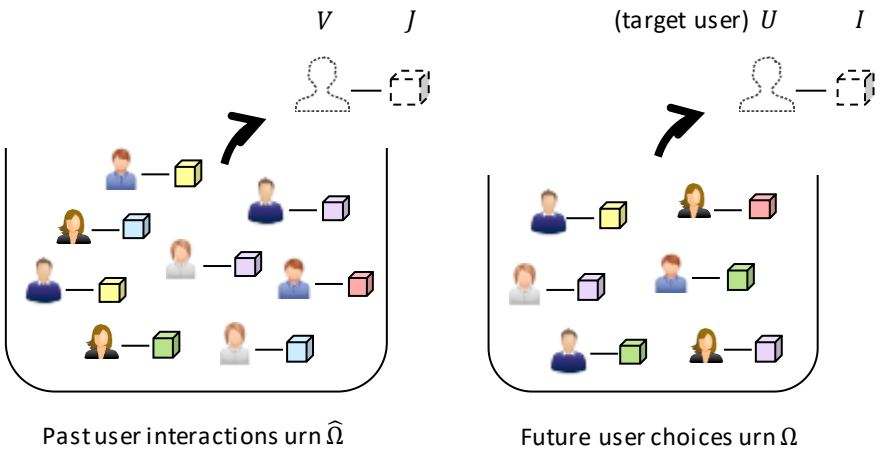

Figure 6.1. Visual representation of the sample space used in the probabilistic kNN formulation.

from the urn according to such probability. Let us call $U: \Omega \to \mathcal{U}$ and $I: \Omega \to \mathcal{I}$ the random variables representing, respectively, the user and the item involved in the selected interaction.

Before continuing with the formalization, we shall point out first an important assumption regarding the way we interpret interactions. Let us consider that at every point in time there is one item that each user likes the most, an item that she would choose above the rest to interact with. We shall assume that interactions faithfully reflect this ideal user-item pair, namely, that users always choose to interact with the item that maximizes their satisfaction.

Using these previous assumptions, we may therefore redefine the recommendation goal as, given a target user, ordering the set of items by the probability that the item will be the one selected by the user to interact with in the next moment. Formally, target items $i \in \mathcal{I}$ must be ranked by the probability $p(I = i|U = u)$ for every target user $u \in \mathcal{U}$. Thus, according to the hypothesis that interactions reflects the selection of the most satisfying option, this ranking must maximize the satisfaction of a user who browses it top-down.

The recommendation problem is therefore reduced to estimating $p(I = i|U = u)$. In the next sections we will derive different versions of kNN by marginalizing such probability by other user (or item) – the neighbor.

6.2.1 User-based kNN

Let us consider now another (this time concrete) set $\widehat{\Omega}$ of all observed past user-item interactions. We similarly represent it as an urn (see Figure 6.1) from which we select another interaction. This time, to explain the probability of each interaction to be selected we consider a point in the past when none of the past interactions have been observed, and for which our past interactions are thus the future ones. At this point, we can consider

the probability of each interaction taking place at the "next moment" and that is the probability we use to sample in the past urn $\widehat{\Omega}$. Let us denote as $V: \widehat{\Omega} \to \mathcal{U}$ and $J: \widehat{\Omega} \to \mathcal{I}$ the user and the item involved in such sampled past interaction, respectively.

Given these two urns, we first estimate $p(I = i|U = u)$ by $p(I = i|U = u, J = I)$. Namely, we assume that the item of both future and past sampled interactions is the same (i.e. $J = I$). In this context the condition $J = I$ is quite vague, since it only states that some undefined user V chose item I at some undefined moment in the past. Thus, we do not expect it largely modifies the original probability, whereas it helps us to establish a connection between V and U: both have rated (or will rate) the same item. Later, this connection will be useful in order to arise to the kNN structure. Now, under the condition $J = I$, we can substitute I by J:

$$p(I = i|U = u) \sim p(I = i|U = u, J = I) = p(J = i|U = u, J = I) \tag{6.5}$$

Applying the law of total sum and marginalizing the previous expression by all the potential values of V – the neighbour – i.e. by all other users, we have:

$$\begin{aligned} & p(I = i|U = u) \\ & \quad \sim p(J = i|U = u, J = I) \\ & \quad = \sum_{v \in \mathcal{U}} p(J = i|V = v, U = u, J = I)\, p(V = v|U = u, J = I) \\ & \quad \sim \sum_{v \in \mathcal{U}} p(J = i|V = v)\, p(V = v|U = u, J = I) \end{aligned} \tag{6.6}$$

In the last step we apply the following independence assumptions to the first term of the sum:

$$p(J = i|V = v, U = u, J = I) \sim p(J = i|V = v, J = I) \sim p(J = i|V = v)$$

Namely, we first remove the condition $U = u$ because we assume that there is not observed interaction between u and i (otherwise i would not be a candidate to be recommended). Then we remove the condition $J = I$ for the same reasons that we use to add it in the first step of Equation 6.5: in absence of more conditions it does not seem to particularly modify the probabilities.

The second term in the sum of Equation 6.6, $p(V = v|U = u, J = I)$, represents the probability that user v interacted in the past with the item that user u has interacted now. This term can be rewritten as:

$$\begin{aligned} p(V = v|U = u, J = I) &= \frac{p(V = v, U = u, J = I)}{p(U = u, J = I)} \\ &= \frac{p(V = v, U = u, J = I)}{\sum_{w \in \mathcal{U}} p(V = w, U = u, J = I)} \end{aligned} \tag{6.7}$$

where we apply first the formula for conditional independence and then the law of total sum for the denominator.

6.2.2 Estimation from observed data

We now use the rating values to estimate the probabilities appearing in Equations 6.6. Thus, we assume that the probability of picking a specific user-item pair when sampling in the past urn $\widehat{\Omega}$ (i.e. the probability of such user-item interaction taking place at a moment in the past) is proportional to the rating value assigned by this user to this item. Namely:

$$p(J = j, V = v) \sim r(v, j) \Big/ \sum_{w \in \mathcal{U}} \sum_{k \in \mathcal{I}} r(w, k)$$

where we divide by the sum of all rating values in order to obtain a correct probability that sums 1 in $\widehat{\Omega}$. Note that if some pair user-item (v, j) do not have a rating present in observed data we simply define $r(v, j) = 0$ in such situation. Thus, directly from previous estimate of $p(J = j, V = v)$ we can provide an expression for the first term of Equation 6.6:

$$p(J = i | V = v) \sim r(v, i) \Big/ \sum_{j \in \mathcal{I}} r(v, j) \tag{6.8}$$

In order to provide an estimate of the second term (developed in Equation 6.7), we first must develop the probability $p(V = v, U = u, J = I)$ as follows:

$$\begin{aligned} p(V = v, U = u, J = I) &= \sum_{j \in \mathcal{I}} p(V = v, U = u, J = I = j) \\ &= \sum_{j \in \mathcal{I}} p(U = u, I = j)\, p(V = v, J = j) \end{aligned}$$

where the last step holds because extracting a pair in the first urn is an independent event from extracting other in the second one.

Now, we approximate the probability distribution $p(U = u, I = j)$ defined over the future urn Ω, by the equivalent distribution defined over the past urn $\widehat{\Omega}$. Namely, $p(U = u, I = j) \sim p(V = u, J = j) \sim r(u, j) / \sum_{w \in \mathcal{U}} \sum_{k \in \mathcal{I}} r(w, k)$. This implies the assumption that preference trends are consistent over time, that they will not vary in the future. This is a strong assumption, but one that is common in non-context-aware recommendation algorithms, and kNN in particular.

Replacing $p(U = u, I = j)$ and $p(V = v, J = j)$ with their corresponding estimates, $p(V = v, U = u, J = I)$ results into:

$$p(V = v, U = u, J = I) \sim \frac{\sum_{j \in \mathcal{I}} r(u, j) r(v, j)}{(\sum_{w \in \mathcal{U}} \sum_{k \in \mathcal{I}} r(w, k))^2}$$

Then, using the development of Equation 6.7, the expression for the second term of Equation 6.6 is given by:

$$p(V = v|U = u, J = I) \sim \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sum_{w \in \mathcal{U}} \sum_{j \in \mathcal{J}} r(u,j) r(w,j)} \tag{6.9}$$

Note that the term $(\sum_{w \in \mathcal{U}} \sum_{k \in \mathcal{J}} r(w,k))^2$ shall appear both in numerator and denominator and have been therefore cancelled. Now, replacing both terms (Equations 6.8 and 6.9) in Equation 6.6 we obtain:

$$p(I = i|U = u) \sim C \sum_{v \in \mathcal{U}} \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sum_{j \in \mathcal{J}} r(v,j)} \, r(v,i)$$

where $C = 1/\left(\sum_{w \in U} \sum_{j \in \mathcal{J}} r(u,j) r(w,j)\right)$ does not depend on i and can be therefore considered as constant given the user u. Consequently, we finally arrive to the following expression to the ranking function:

$$p(I = i|U = u) \propto \sum_{v \in \mathcal{U}} \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sum_{j \in \mathcal{J}} r(v,j)} \, r(v,i) \tag{6.10}$$

If we compare previous expression with Equation 6.1, we can see that it defines a user-based kNN where the similarity function is given by:

$$sim(u,v) = \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sum_{j \in \mathcal{J}} r(v,j)} = p(U = u, J = I|V = v) \tag{6.11}$$

In fact, we can add the term $\sum_{j \in \mathcal{J}} r(u,j)$ to the denominator of the similarity and the resulting ranking will be exactly the same, since this term does not depend on i. By adding such term, we would obtain a similarity that looks quite like a cosine (as in Equation 6.3), only using L_1 norm instead of L_2. We have thus managed to formulate the ranking function of a common user-based kNN as a probability. In particular, we have developed a non-normalized variant, since it is equivalent to Equation 6.1 with $C = 1$.

We may now want to restrict the sum over all users to a subset of neighbours of u ($N_k[u]$), as happens in Equation 6.1. Formally, this implies adding a new condition to the original probability, $p(I = i|U = u) \sim p(I = i|U = u, V \in N_k[u])$ and, consequently, to all the probabilities that appear in Equation 6.6.

$$p(I = i|U = u)$$
$$\sim \sum_{v \in N_k[u]} p(J = i|V = v, V \in N_k[u]) \, p(V = v|U = u, J = I, V \in N_k[u])$$

In practice, in the previous mathematical development we only need to change the subscripts of the sums over users, i.e. change $v/w \in \mathcal{U}$ by $v/w \in N_k[u]$. Thus, the final ranking function defined by Equation 6.10 remains the same but summing on $v \in N_k[u]$

instead of on $v \in \mathcal{U}$. The value $C = 1/\sum_{w \in N_k[u]} \sum_{j \in \mathcal{J}} r(u,j) r(w,j)$ is still constant on i and can be therefore removed again:

$$p(I = i | U = u) \propto \sum_{v \in N_k[u]} \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sum_{j \in \mathcal{J}} r(v,j)} \, r(v,i)$$

In our experiments, similarly to heuristic kNN, we shall consider neighbour selection based on the highest values of the similarity $sim(u,v)$, which in our case corresponds to $p(U = u, J = I | V = v)$ (see Equation 6.11).

6.2.3 Normalized variant

Just as we restricted the set of users v to $N_k[u]$, we may consider any other subset of $\mathcal{U}$. In particular, we can restrict V to those users that have been seen interacting with i: formally $p(I = i | U = u) \sim p(I = i | U = u, r(V,i) \neq \emptyset)$. Modifying the sums over users of Equation 6.10 to include this new constraint we obtain the following expression:

$$\begin{aligned} & p(I = i | U = u) \\ & \sim \sum_{\substack{v \in \mathcal{U} \\ r(v,i) \neq \emptyset}} p(J = i | V = v, r(V,i) \neq \emptyset) \, p(V = v | U = u, J = I, r(V,i) \neq \emptyset) \\ & \sim C \sum_{v \in \mathcal{U}} \frac{\sum_{j \in \mathcal{J}} r(u,j) r(v,j)}{\sum_{j \in \mathcal{J}} r(v,j)} \, r(v,i) \end{aligned} \tag{6.12}$$

where $C = 1/\sum_{w \in \mathcal{U}: r(w,i) \neq \emptyset} \sum_{j \in \mathcal{J}} r(u,j) r(w,j)$. In the last step we remove the constraint $r(v,i) \neq \emptyset$ in the subscript of the external sum since it does not alter the result, i.e. it is implicit in the product by $r(v,i)$ which is 0 is v have not rated the item i. Note that this time we cannot remove constant C because it depends on item i.

Now, the numerator of the similarities (the term $\sum_{j \in \mathcal{J}} r(u,j) r(v,j)$) divided by C sums 1 over different neighbors, similar in structure to the heuristic normalized kNN (except for the term $\sum_{j \in \mathcal{J}} r(v,j)$ in the denominator). Neighbor selection can be introduced in a similar way as in the non-normalized variant, simply replacing the subscripts $v/w \in \mathcal{U}$ by $v/w \in N_k[u]$.

6.2.4 Item-based kNN

In order to obtain an item-based development of the term $p(I = i | U = u)$, the previous analysis can be conducted following an item-oriented approach. For that purpose, we shall start by inverting the user and the item in the initial ranking function:

$$p(I = i | U = u) = p(I = i) \frac{p(U = u | I = i)}{p(U = u)} \propto_u p(I = i) \, p(U = u | I = i)$$

In the last step we remove the term $p(U=u)$ since it is constant over the different items and thus does not affect the ranking.

Now we address the formalization of the term $p(U=u|I=i)$ in a similar way that we did with $p(I=i|U=u)$ in the user-based version. First, we assume that the user involved in both future and past sampled interactions is the same. Namely, we add the condition $V=U$. Again, we consider that such event $V=U$, that states the user u has rated another undefined item at some point in the past, does not provide meaningful information and therefore may not significantly modify the probability $p(I=i|U=u)$. Under the condition $V=U$, we can substitute U by V:

$$p(U=u|I=i) \sim p(U=u|I=i,V=U) \sim p(V=u|I=i,V=U)$$

Now marginalizing $p(V=u|I=i,V=U)$ by all the potential neighbours j, and applying similar independence assumptions as in the user-based version we obtain:

$$\begin{aligned} &p(I=i|U=u) \\ &\quad \sim p(I=i)\, p(V=u|I=i,V=U) \\ &\quad = p(I=i) \sum_{j\in\mathcal{I}} p(V=u|\,J=j,I=i,V=U)\, p(J=j|\,I=i,V=U) \\ &\quad \sim p(I=i) \sum_{j\in\mathcal{I}} p(V=u|\,J=j)\, p(J=j|\,I=i,V=U) \end{aligned} \tag{6.13}$$

Estimating each of the probabilities appearing in previous expression from observed ratings – in the same way we documented in Section 6.2.2 – we get:

$$p(I=i) = \sum_{w\in\mathcal{U}} r(w,i) \Big/ \sum_{w\in\mathcal{U}} \sum_{k\in\mathcal{I}} r(w,k)$$

$$p(V=u|\,J=j) = r(u,j) \Big/ \sum_{v\in\mathcal{U}} r(v,j)$$

$$p(J=j|\,I=i,V=U) = \frac{p(J=j,I=i,V=U)}{\sum_{k\in\mathcal{I}} p(J=k,I=i,V=U)} = \frac{\sum_{v\in\mathcal{U}} r(v,i)r(v,j)}{\sum_{k\in\mathcal{I}} \sum_{v\in\mathcal{U}} r(v,i)r(v,k)}$$

And then, the final expression for $p(I=i|U=u)$ using this item-based variant is:

$$\begin{aligned} p(I=i|U=u) &\propto \left(\sum_{v\in\mathcal{U}} r(v,i)\right) \sum_{\substack{j\in\mathcal{I} \\ j\neq i}} \frac{\sum_{v\in\mathcal{U}} r(v,i)r(v,j)}{\sum_{k\in\mathcal{I}} \sum_{v\in\mathcal{U}} r(v,i)r(v,k)} \cdot \frac{r(u,j)}{\sum_{v\in\mathcal{U}} r(v,j)} \\ &\propto C \sum_{j\in\mathcal{I}} \frac{\sum_{v\in\mathcal{U}} r(v,i)r(v,j)}{\sum_{v\in\mathcal{U}} r(v,j)} r(u,j) \end{aligned} \tag{6.14}$$

with $C = \sum_{v\in\mathcal{U}} r(v,i) / \sum_{j\in\mathcal{I}} \sum_{v\in\mathcal{U}} r(v,i)r(v,j)$.

Now we see that this formulation presents analogies with the heuristic item-based kNN (Equation 6.2) if we consider the following similarity.

$$sim(i,j) = \frac{\sum_{v\in\mathcal{U}} r(v,i)r(v,j)}{\sum_{v\in\mathcal{U}} r(v,j)} = p(I=i, V=U|J=j)$$

One difference with cosine similarity is that now we cannot add the term $\sum_{v\in\mathcal{U}} r(v,i)$ to the denominator because it depends on the item i. Such term is actually multiplying the whole expression, since it appears in the numerator of C. However, the denominator $\sum_{j\in\mathcal{J}}\sum_{v\in\mathcal{U}} r(v,i)r(v,j)$ is compensating this difference to some extent.

As in the user-based variant we may restrict the sums over items to some specific subset, as the closest neighbours of i and/or those items already rated by target user u. This will result in a modification of the sum subscripts, replacing $j \in \mathcal{J}$ by $j \in \mathrm{N_k}[i]$ or $j \in \mathcal{J}: r(u,j) \neq \emptyset$, respectively. Or combining both restrictions if we so wish: $j \in \mathrm{N_k}[i] : r(u,j) \neq \emptyset$. Note that these modifications also affect the constant C. In fact, the restriction to already rated items (by target user u) gives rise a normalized variant of the Equation 6.14 – as it happened with the user-based kNN version – precisely because the numerator of the similarities divided by the denominator of constant C sums 1 over items: $C = \sum_{v\in\mathcal{U}} r(v,i) / \sum_{j\in\mathcal{J}: r(u,j)\neq\emptyset}\sum_{v\in\mathcal{U}} r(v,i)r(v,j)$.

6.2.5 Smoothing

We found in our experiments that smoothing the probability estimates of the similarity functions slightly improves empirical results. Specifically, for user-based version we tested Bayesian smoothing with Dirichlet prior (Büttcher et al. 2010) on the terms described by Equation 6.11, i.e. the probabilities $p(U=u, J=I|V=v)$. For this probability the Dirichlet prior is $p(U=u, J=I) \propto \sum_{w\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(u,j)r(w,j) \,/\, \sum_{w\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(w,j)$, so the final estimate results in:

$$\begin{aligned} p(U=u, J=I|V=v) &\sim \frac{\sum_{j\in\mathcal{J}} r(u,j)r(v,j) + \mu'\, p(U=u, J=I)}{\sum_{j\in\mathcal{J}} r(v,j) + \mu'} \\ &\propto \frac{\sum_{j\in\mathcal{J}} r(u,j)r(v,j) + \mu \dfrac{\sum_{w\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(u,j)r(w,j)}{\sum_{w\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(w,j)}}{\sum_{j\in\mathcal{J}} r(v,j) + \mu} \end{aligned} \tag{6.15}$$

where we redefine $\mu = \mu' / \sum_{w\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(w,j)$. This smoothing applies both to the normalized and the non-normalized variants, since both use the probability $p(U=u, J=I|V=v)$ as similarity function and neighbour selection criterion.

Similarly, in the case of item-based variants we smoothed the analogous probability $p(I=i, V=U|J=j)$ using $p(I=i, V=U)$ as prior estimate:

$$p(I=i,V=U|J=j) \sim \frac{\sum_{v\in\mathcal{U}} r(v,i)r(v,j) + \mu \frac{\sum_{j\in\mathcal{J}}\sum_{v\in\mathcal{U}} r(v,i)r(v,j)}{\sum_{w\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(w,j)}}{\sum_{v\in\mathcal{U}} r(v,j) + \mu} \quad (6.16)$$

In the experiments of Section 6.4.2 we will empirically analyse the effect of varying the smoothing parameter μ, as well as propose and justify taking as default value $\mu = \frac{1}{|\mathcal{U}|}\sum_{v\in\mathcal{U}}\sum_{j\in\mathcal{J}} r(v,j)$.

6.3 Popularity bias and the neighbour hypothesis

As we already mentioned in Section 6.1, the main intuition underlining kNN formulation is that similar users in the past will remain being similar in the future. This consistency over time is needed by essentially any recommendation approach, since all of them use past user interactions to predict future interests. However, hidden in previous statement is one more subtle assumption, that for each user there are indeed some users more similar than others, that there is some level of dependency or clustering between users. Otherwise, similarity would be as informative as random association.

In this section we formally prove that when this assumption is not met, namely, when we have user independence, the different kNN variants are reduced to popularity-based recommendations. This reveals the connection between popularity and kNN we were looking for.

6.3.1 User-based bias

The user independence assumption formally means that $p(V=v|U=u,J=I) \sim p(V=v)$ for all v and u. Let us see why it is so before studying how it affects to the behaviour of the user-based kNN variant. According to the estimations of Section 6.2.2, the previous probabilities present the following expressions:

$$p(V=v|U=u,J=I) \propto \sum_{j\in\mathcal{J}} r(u,j)r(v,j)$$

$$p(V=v) \propto \sum_{j\in\mathcal{J}} r(v,j)$$

Therefore, the equivalence between both values implies that the intersection of v and u is proportional to the ratings of v. In other words, that there is not any special connection between both users – that they do not tend to rate the same or distinct items. Their preferences coincide to the same extent they would under a random rating assignment. If this happens for all v and u pairs, then there is not any level of clustering or user dependency – the situation we want to represent.

Now, introducing the assumption $p(V = v|U = u, J = I) \sim p(V = v)$ in Equation 6.6 results in:

$$p(I = i|U = u) \sim \sum_{v \in \mathcal{U}} p(J = i|V = v)\, p(V = v) = p(J = i)$$

Thus, under this assumption we are approximating the ranking function $p(I = i|U = u)$ by the probability that a random user picks the item i, $p(J = i)$, which may be indeed understood as a natural notion of item popularity. This means that popularity represents a more inexact approximation of $p(I = i|U = u)$ than kNN, inasmuch as the former relies on an extra independence assumption. However, to the extent that $p(V = v|U = u, J = I)$ becomes independent from the user u and gets closer to $p(V = v)$, popularity would be a better approximation of $p(I = i|U = u)$. In other words, kNN reduces to popularity-based recommendation in the absence of any dependence between the user preferences.

On the contrary, when the conditional distribution $p(V = v|U = u, J = I)$ diverges from the neighbour prior $p(V = v)$, popularity becomes a worse approximation of $p(I = i|U = u)$. In such situations, it is expected that kNN performs better, since it presents a more exact development of the previous probability.

Regarding the specific value for the probability $p(J = i)$, it may be estimated from observed data applying equivalent estimations to the ones of Section 6.2.2 as:

$$p(J = i) \sim \frac{\sum_{v \in \mathcal{U}} r(v, i)}{\sum_{w \in \mathcal{U}} \sum_{j \in \mathcal{I}} r(w, j)} \propto \sum_{v \in \mathcal{U}} r(v, i)$$

In other words, we estimate $p(J = i)$ as the sum of ratings of i. Note that this is not exactly the definition of popularity that we are handling since we are considering the number of ratings and not the sum. However, it seems sensible to assume that both values present a high correlation and that, therefore, ranking by one is practically equivalent to ranking by the other. Actually, that is exactly the case if we consider binary relevance, since then the sum of ratings of an item is the same as its relevant popularity.

To sum up, the probabilistic formulation evidences that there is a structural connection between the user-based non-normalized kNN variant and relevant popularity, and justifies why pairwise user dependence is a determining factor for kNN to perform better.

6.3.2 Normalized variant bias

In the scope of the normalized variant, the pairwise user independence means that $p(V = v|U = u, J = I, r(V, i) \neq \emptyset) \sim p(V = v|\, r(V, i) \neq \emptyset)$ for all v and u. In other words, we only worry about the dependences between the users that take part in the probabilistic development of the normalized variant, i.e. those that have rated i. The behaviour of the rest of users is not considered by such version of kNN, and therefore does not have influence in the result.

Introducing the user independence condition in Equation 6.12, we get:

$$p(I = i|U = u) \sim \sum_{v \in \mathcal{U}} p(J = i|V = v, r(V,i) \neq \emptyset)\, p(V = v|\, r(V,i) \neq \emptyset)$$
$$= \sum_{v \in \mathcal{U}} p(J = i, V = v|\, r(V,i) \neq \emptyset)$$
$$\sim \sum_{v \in \mathcal{U}} r(v,i) \Big/ \sum_{\substack{v \in \mathcal{U} \\ r(v,i) \neq \emptyset}} \sum_{j \in \mathcal{I}} r(v,j)$$

Thus, this time the relevant popularity of the item – represented by $\sum_{v \in \mathcal{U}} r(v,i)$ – is divided by an extra term $\sum_{v \in \mathcal{U}: r(v,i) \neq \emptyset} \sum_{j \in \mathcal{I}} r(v,j)$. If each item has been rated by highly and moderately active users to the same extent, i.e. the terms $\sum_{j \in \mathcal{I}} r(v,j)$ are not too different from one item to other, then $\sum_{v \in \mathcal{U}: r(v,i) \neq \emptyset} \sum_{j \in \mathcal{I}} r(v,j)$ is approximately proportional to the number of users who have rated the item i. That means that $p(I = i|U = u)$ is equivalent to the quotient between relevant popularity and total popularity, namely, it is average rating. In order words, when we add the condition $r(V,i) \neq \emptyset$ to the non-normalized variant we are roughly dividing it by the total popularity of the item (the number of users V that have rated it) , moving it closer to average rating.

Therefore, we reach an important finding here: the normalized and non-normalized variants present fundamental different behaviours, the former lies in the distribution of average rating whereas the latter is supported by relevant popularity. Moreover, we see in Chapter 4 that relevant popularity is unfairly rewarded above average rating in common offline experiments. This might explain why normalized variants typically perform worse that non-normalized ones and, what is more important, it could be suggesting that the real and true sign of the comparison might be exactly the opposite. The experiment of Chapter 4 (Section 4.8.3), is a first sign in this line, as it displays a situation where the normalized user-based kNN is above the non-normalized one when we compute true metric values, but bellow if we only look at the observed values.

6.3.3 Item-based bias

In the item-oriented approach, the independence condition is translated to items as follows: $p(J = j|I = j, V = U) \sim p(J = j)$. Under this assumption, Equation 6.13 becomes:

$$p(I = i|U = u) \sim p(I = i) \sum_{j \in \mathcal{I}} p(V = u|\, J = j) p(J = j) = p(I = i)$$

Thus, similarly to the user-based version, item-based kNN also degrades to relevant popularity when the conditional distribution $p(I = i|J = i, V = U)$ gets closer to the prior $p(I = i)$.

Regarding the normalized item-based variant, we assume the independence condition $p(J = j|I = j, V = U, r(u,J) \neq \emptyset) \sim p(J = j|\, r(u,J) \neq \emptyset)$. Then, the resulting ranking function results in:

$$p(I = i|U = u) \sim p(I = i) \sum_{j \in \mathcal{J}} p(V = u|\, J = j, r(u,J) \neq \emptyset)\, p(J = j|\, r(u,J) \neq \emptyset)$$

$$\propto p(I = i)$$

where the last step holds because the sum does not depend on i and can be therefore removed. Thus, in this case we obtain that the normalized item-based variant is related to relevant popularity, and not to the average rating as its normalized user-based counterpart.

6.4 Empirical observation

In order to verify the previous trends and analytical results, we now run a series of experiments on MovieLens, Netflix and Last.fm datasets. Datasets that we have been using along this thesis to represent common publicly available data and whose dimensions can be found in Table 2.1 (Chapter 2). Later in this section we will also use the CM100k dataset described in Chapter 4 (Section 4.8.1) to test the behaviour of the algorithms in absence of discovery and rating biases.

6.4.1 General performance

The first verification of all the previous results that we shall conduct is whether the probabilistic kNN variants are really representative of the heuristic ones, particularly whether they obtain an equivalent performance. In order to do that, we implement the non-normalized probabilistic user-based (PUB) and item-based (PIB) variants, as well as their corresponding normalized versions (nPUB and nPIB, respectively). For the heuristic kNN algorithms we take the implementations of the non-normalized variants – both user-based (HUB) and item-based (HUB) – provided by the public library RankSys[5]. Such library does however not include non-normalized adaptations, so we also implement them (nHUB and nHIB for the user-based and item-based approaches, respectively). In all the previous heuristic variants we use cosine similarity (see Equations 6.3 and 6.4).

In total, between the probabilistic and heuristic modalities, we have eight kNN variants to compare. Note that the neighbourhood size (denoted by k) is a configuration parameter in all of them, so we select the best value (in terms of P@10) of this parameter for each version. For that purpose, we conduct a grid search starting with steps of 10 in the 10-100 interval, then steps of 100 in 100-1,000, and so forth. Table 6.3 shows the

[5] http://ranksys.org

	HUB	PUB	HIB	PIB	nHUB	nPUB	nHIB	nPIB
MovieLens	50	50	100	100	10	20	10	40
Netflix	100	100	100	50	10	10	10	100
Last.fm	100	500	∞	∞	10	10	10	∞

Table 6.1. Neighborhood size k in the kNN configuration on each dataset ($k = \infty$ indicates all items are taken as neighbors).

	PUB	PIB	nPUB	nPIB
MovieLens	90	200	90	1,000
Netflix	200	20,000	200	20,000

Table 6.2. Smoothing parameter μ for each probabilistic kNN variant on each dataset.

resulting neighbourhood size for each kNN variant on each dataset. For all the normalized variants we require a minimum of 5 neighbour ratings for an item in order to be recommended, otherwise these variants may suffer the same bias than average rating towards items with very few – but relevant – ratings.

We found that smoothing the similarity function, as we formalized in Section 6.2.5, slightly improves the probabilistic algorithms on Netflix and MovieLens. Table 6.4 displays the values for parameter μ (see Equations 6.15 and 6.16) that optimize (again in terms of P@10 and with a grid search) the smoothing of each probabilistic version on each dataset. We do not include Last.fm since in such dataset the smoothing does not significantly modify any probabilistic variant. We will analyse in more detail the effect of such smoothing later in Section 6.4.2, where we will also explain how to select a default value for this parameter.

We shall also preprocess the rating values on MovieLens and Netflix to avoid negative ratings increasing the sums of kNN, as if they indicated positive preference. Otherwise, and item with a lot of negative neighbour ratings might obtain a higher score than other with less but positive ratings. Therefore, we map negative values (values 1, 2 and 3) to 0, ratings with value 4 to 1 and ratings with value 5 to 2. This is not necessary on Last.fm, since it is a register of music playcounts and does therefore not contain negative preferences.

As a frame of reference, in this comparison we also include non-personalized recommendation algorithms: random recommendation, relevant popularity and average rating (numeric, not binary). Similar to the case of normalized kNN variants, in the recommendation by average rating we only consider as candidates those items with more than 5 ratings. Finally, we include matrix factorization as a top-performing reference. We take the implementation provided by RankSys with $k = 20$ factors, $\alpha = 1$, and $\lambda = 0.1$. Regarding the number of iterations, we take 20 iterations on MovieLens and Last.fm, and

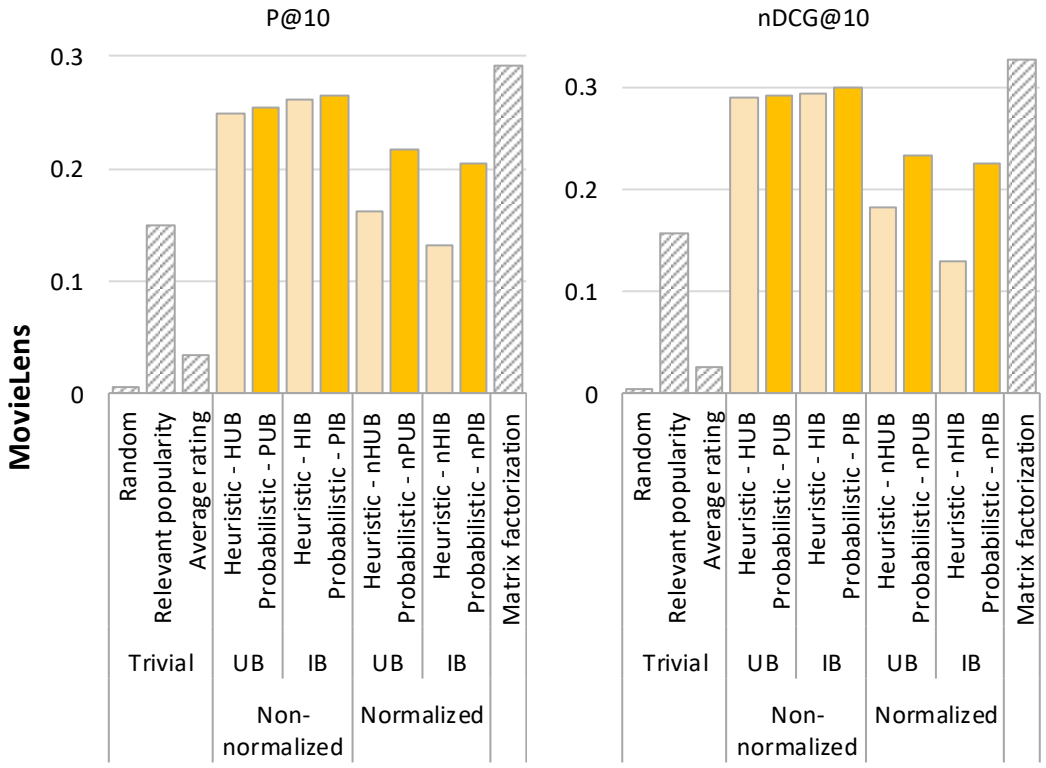

Figure 6.2. Comparative performance of all kNN variants on MovieLens.

50 iterations on Netflix. These are the same parameter configurations that we employed in the experiments of Chapter 2, when we provided context for this thesis.

To compute the performance of the previous algorithms we divide the rating data in training and test sets with a split ratio of $\rho = 0.8$, as in the preceding chapters. Figures 6.2 (MovieLens) and 6.3 (Netflix and Last.fm) display such performance in terms of P@10 and nDCG@10. The bars representing equivalent heuristic and probabilistic versions are depicted next to each other for ease of comparison. We use a darker colour for the probabilistic versions, and a streaked grey pattern for the non-personalized algorithms and matrix factorization.

We can see that the probabilistic versions present quite a similar performance to the corresponding heuristic ones, especially in the non-normalized approach (i.e. PUB vs. HUB and PIB vs. HIB). In MovieLens and Netflix, PUB and PIB perform slightly better that their heuristic counterparts. In Last.fm PIB is also above HIB (this time with a greater margin), but PUB is surpassed by HUB.

The difference, however, becomes greater when we move to the normalized versions (nPUB vs. nHUB and nPIB vs. nHIB). In this case, most of the probabilistic variants overcome the heuristic ones (except for nPUB and nHUB on Last.fm) by a considerable margin. Note also that normalized approaches perform systematically worse than the non-normalized ones (again with the exception of nHUB and HUB on Last.fm) as is common

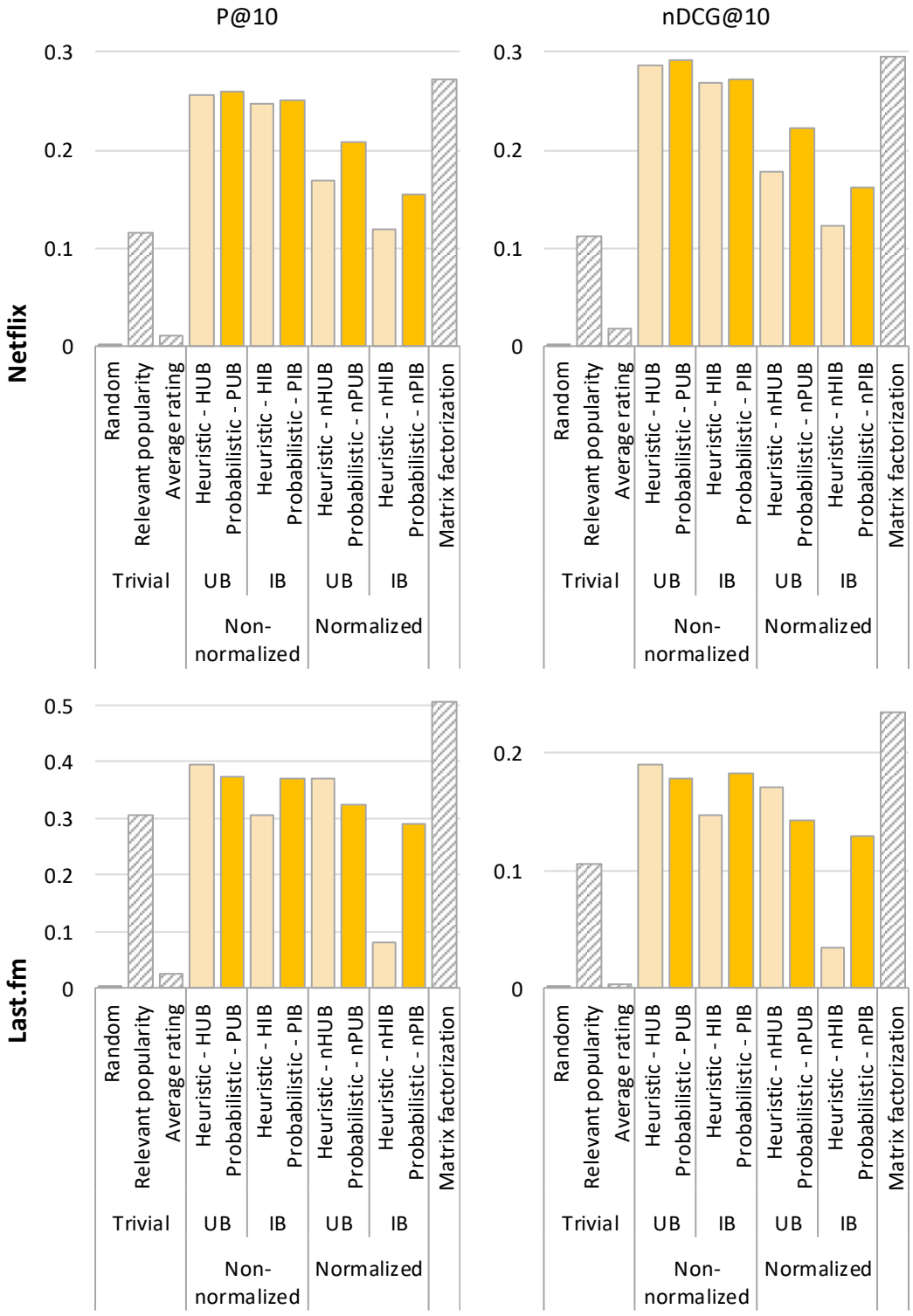

Figure 6.3. Comparative performance of all kNN variants on Netflix and Last.fm.

in this kind of offline experiments. This might be explained by the lower performance of average rating compared to popularity.

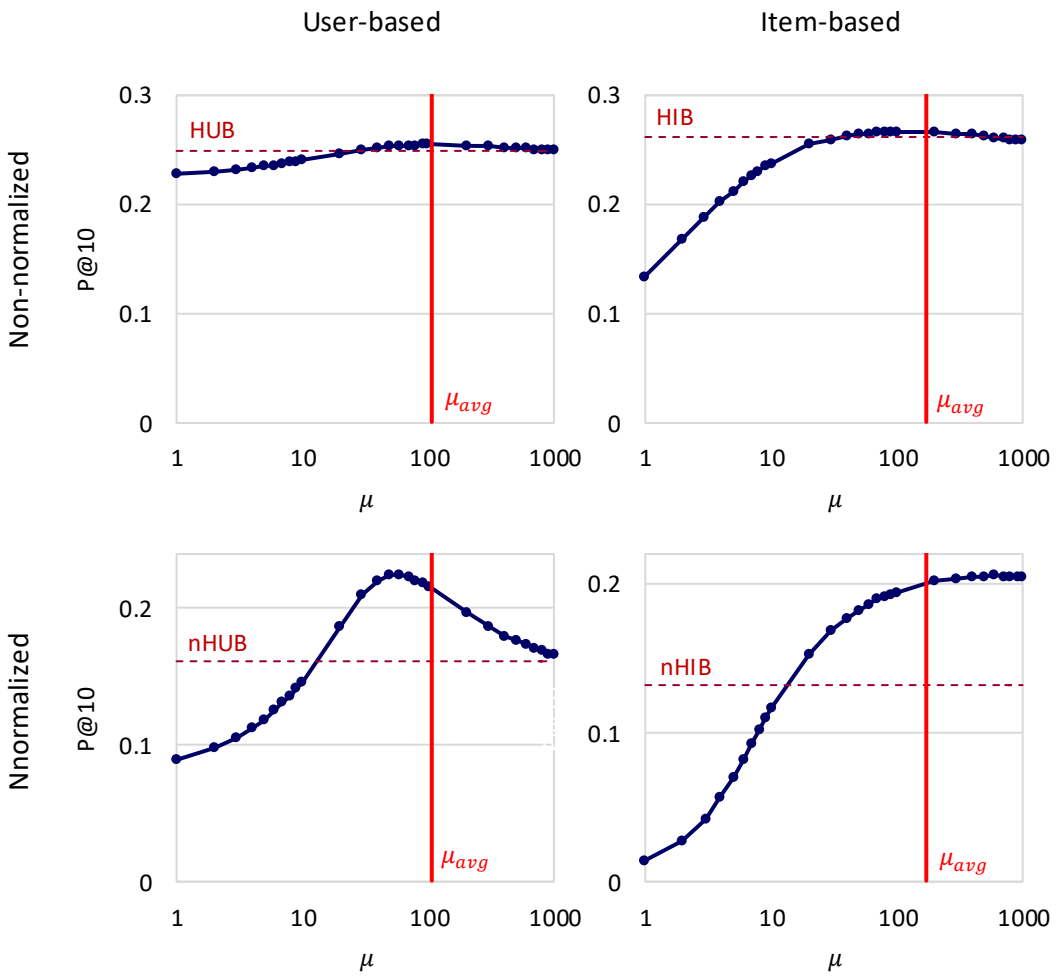

Figure 6.4. Performance of the probabilistic kNN variants – PUB (top-left), PIB (top-right), nHUB (bottom-left), nHIB (bottom-right) – for different values of the smoothing parameter μ.

Finally, the best-performing algorithm is matrix factorization, although not by a great margin with respect to the best kNN algorithm, particularly on Netflix.

6.4.2 Smoothing

As with the neighbourhood size parameter (k), we also carry out a grid search to find the optimal order of the smoothing parameter μ (see Equations 6.15 and 6.16). However, in this case such parameter is not delimited and might be as large as we want (k is limited by number of users). Actually, consulting Table 6.4 with the optimal values of such parameter on MovieLens and Netflix we observe an incredible wide variation, from 90 in the user-based variant on MovieLens to 20,000 in the item-based approach on Netflix.

This parameter sweep can thus be a time-wasting and costly procedure, especially for those experiments where kNN is not the main algorithm but a reference point or baseline.

Therefore, it would be convenient to have a default value that, even if it is not the exact optimal, is close to it and serves as a suggestion of the range we must sweep to find such optimal value.

In Bayesian smoothing with Dirichlet priors, a reasonable and typical default value for μ that produces near-optimal results is the expected value of the smoothed quotient denominator (Büttcher et al. 2010). In our case, we are smoothing the probability $p(U = u, J = I|V = v)$ – the similarity function – whose denominator is $p(V = v)$, so we suggest using $\mu_{avg} = \frac{1}{|\mathcal{U}|}\sum_{v\in\mathcal{U}}\sum_{j\in\mathcal{I}} r(v,j)$ as a default value. In other words, the average number (sum) of ratings per user.

Figure 6.4 shows the effect of varying μ parameter in the performance of each probabilistic variant on MovieLens. As reference point the accuracy of the heuristic counterpart is indicated with a horizontal line. Moreover, we highlight with a red vertical line the performance obtained by the default value we propose above. One first comment to make observing the graphs is that, in all cases, the smoothing drastically improves the performance of the non-smoothed option (except for the non-normalized user-based PUB). Moreover, such non-smoothed option is indeed always below the heuristic variant. The reason of this behaviour is that the different between both probabilistic and heuristic variants is roughly that the former uses L_1 norm in the user similarity whereas the later users L_2. And in fact, L_2 norm, as far as it operates with root squares, acts as a smoothing of L_1 norm. Another aspect to point out is that the default value we propose por the smoothing parameter μ – denoted with μ_{avg} in Figure 6.4 – is quite close to the optimal value in all cases.

6.4.3 Popularity biases

We empirically address now the question as to what extent kNN introduces a popularity component in its recommendations. We already advanced some results in this line in Chapter 2 (Section 2.4) where we represented the number of times each item is recommended versus its popularity, for three different algorithms: the heuristic non-normalized user-based kNN, matrix factorization and the optimal personalized recommendation. Now we extend such study for the rest of kNN variants, including the probabilistic ones, and consider also the bias towards average rating. Thus, Figure 6.5 displays, for each (probabilistic and heuristic) user-based kNN variant, the number of times that each item appears in the top 10 positions of the ranking (note that there is one ranking per user) versus its number of relevant ratings (top) and its average rating (bottom), using MovieLens dataset. If an item has not been recommended for any user, it is excluded from the graph, namely, we have removed the points whose value in the y coordinate is 0. Regarding the neighbourhood size of each algorithm, we take all users as neighbours to avoid any possible distorting effect. For the same reason, we do not consider any smoothing for the probabilistic variants.

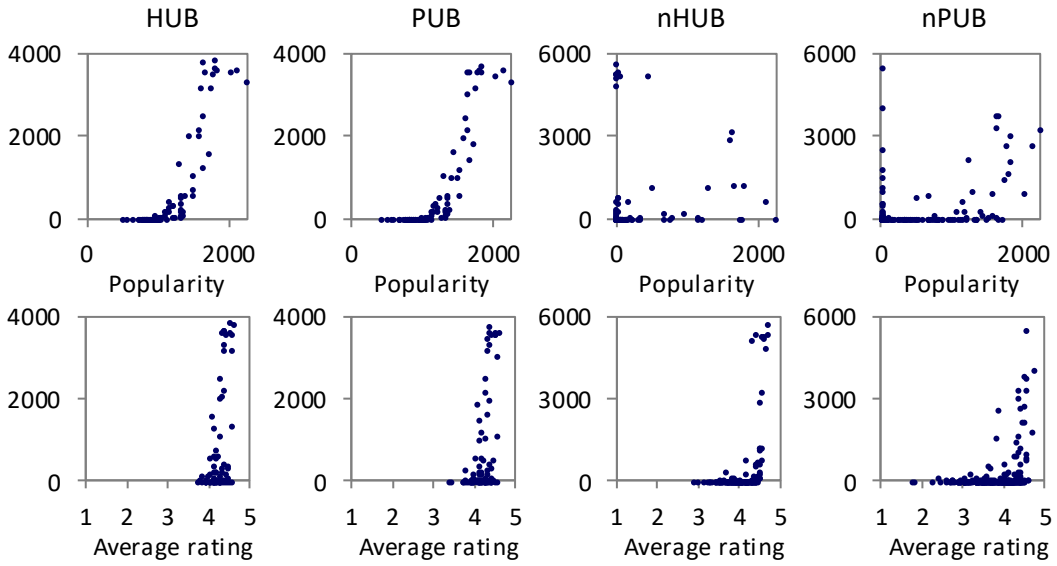

Figure 6.5. Bias towards relevant popularity (top) and average rating (bottom) of the heuristic and probabilistic user-based kNN variants on MovieLens. Each point in the plots represents an item, the x axis shows the number of relevant ratings of the item, and the y coordinate is the number of times (i.e. the number of users to whom) the item is recommended in the top 10 by the corresponding algorithm.

We can see that both HUB and PUB present a clear bias towards popular items, confirming analytical trends of Section 6.3.1. In fact, the top most popular items are the most recommended by these two non-normalized user-based variants. Regarding the comparison with average ratings, while it is true that most recommended items (those with the largest y coordinate) present a high average rating superior to 4, the opposite is not true, namely, the items with the largest average rating are not necessarily the most recommended (remember that items with a 0 in the y coordinate are not shown). Moreover, the high average rating presenting by most recommended items is possibly due to the fat that on MovieLens, the items with the largest number of relevant ratings present also a high average rating (more than 4), as we can verify in Figure 6.8. In other words, that the obvious bias that these two algorithms have towards popularity is most probably the cause of the vague trend we also observe with average rating.

We also note that both HUB and PUB present quite a similar shape, a further argument in support of the probabilistic version being a representative variant of the kNN approach. This similarity is also notable between nHUB and nPUB. Indeed, the bias towards popular items completely disappears in both normalized variants, whereas the bias towards average rating becomes more evident (the point cloud of most recommended items has moved to the right). Now the items with the largest average rating are quite recommended. These results confirm once again the analytical trends we have derived in Section 6.2.3.

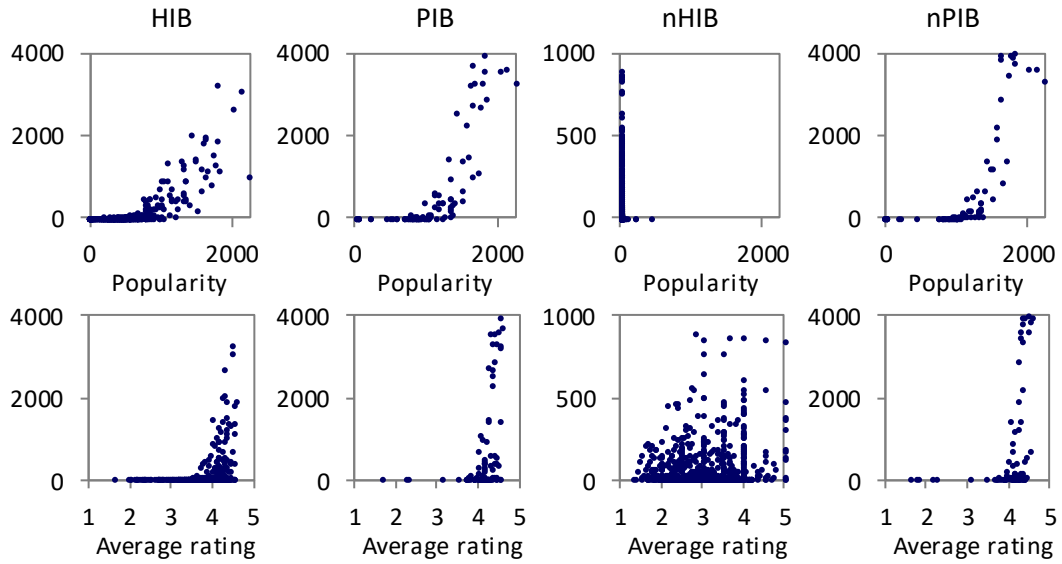

Figure 6.6. Bias towards relevant popularity (top) and average rating (bottom) of the heuristic and probabilistic item-based kNN variants on MovieLens. The x and y axes have the same meaning as in Figure 6.5.

Let us move now to the item-based variants, whose bias towards popularity and average rating on MovieLens is depicted in Figure 6.6. We can see that the behaviour of the non-normalized versions, HIB and PIB, is quite like the one of their user-based counterparts, namely, the popularity bias is also quite strong (particularly in the heuristic variant) and the shape of the bias towards average rating is practically the same. However, the differences arise when we move to normalized approaches. Thus, the probabilistic one (nPIB) is similar in structure to its non-normalized version (PIB), i.e. it is biased towards popularity instead towards average rating, as was the case of the equivalent user-based algorithms. On the contrary, the heuristic normalization nHIB presents a bias against popularity, to the extent that the recommended items are precisely those with less relevant ratings. Indeed, the items with more than 500 rating are fully ignored. This might explain somehow the poor performance achieved by nHIB on MovieLens, Netflix and Last.fm (see Figures 6.2 and 6.3). The trend towards average rating also disappears, and it is difficult to find any other bias in this heuristic version.

6.4.4 Performance in absence of biases

Let see now what happen when we remove the discovery and rating decision biases that most likely interfere in the rating generation process of common available data, as MovieLens, Netflix and Last.fm datasets. There is a possibility that such biases could be artificially increasing the user dependencies, beyond the ones purely caused by different user tastes. For instance, users targeted by the same marketing campaigns would be impelled to discover the same items and, therefore, to rate them. In such situation, kNN might be

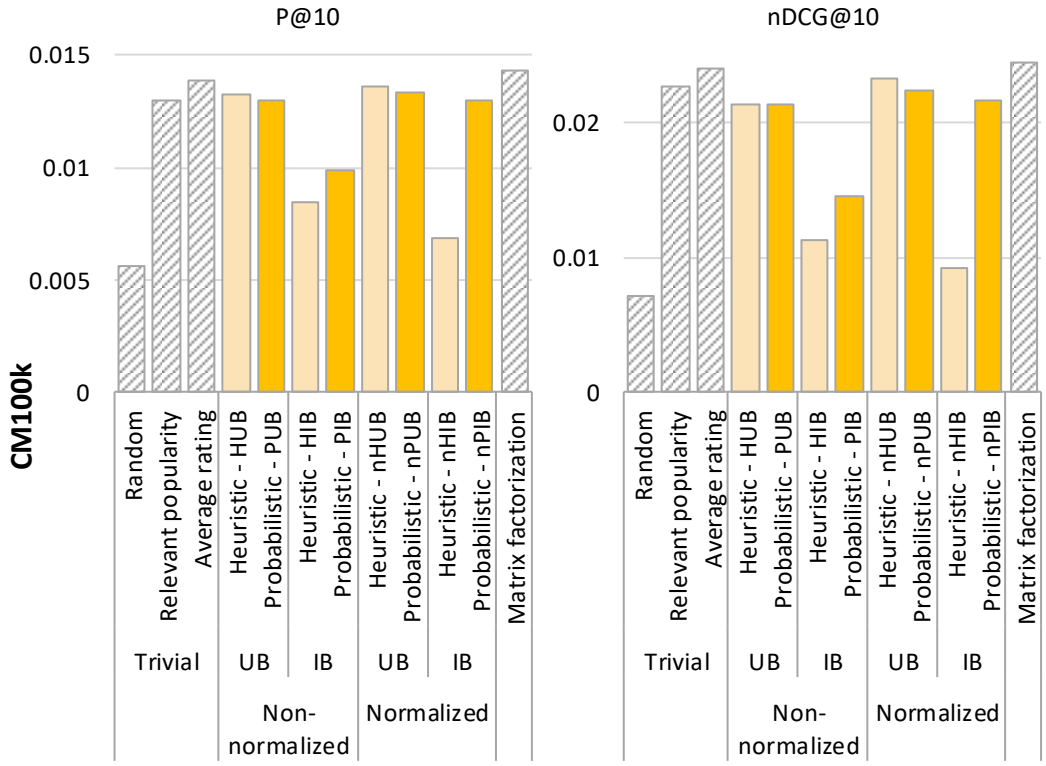

Figure 6.7. Comparative performance of all kNN variants on CM100k.

leaded by discovery patterns instead of by real preferences, and thus the performance we are observing in Figures 6.2 and 6.3 might not be real.

Consequently, we now repeat the performance analysis and popularity bias study of the previous sections, but using as input data the CM100k dataset explained in Chapter 4. In this case, no smoothing or minimum is required for normalized kNN versions and average rating, since on CM100k all items present roughly the same number of ratings (see Figure 4.9 of Chapter 4). The optimal neighbourhood size for all the kNN variants is $k = \infty$ (i.e. take all users as neighbours), except for nHIB where we take $k = 200$. We configure matrix factorization with $k = 5$, $\alpha = 10$, $\lambda = 500$, and 20 iterations. Finally, due to the reduced dimensions of this dataset we repeat the experiment 10 times to smooth variance.

Figure 6.7 displays the performance of the algorithms on CM100k, where all judgments are taken as input data for recommendation algorithms, namely, without differentiating between discovered and non-discovered items (case a of Table 4.1). The first thing to note is that the relative performance of all algorithms has substantially decreased compared to the MovieLens dataset, to the point that the best algorithm (matrix factorization) is only three times better than random. This can be explained by the flat rating distribution of this dataset, which is caused in turn by the absence of discovery and rating decision biases. Remember from Chapter 4 that in this scenario the optimal non-personalized recommendation for observed precision ranks items according to $p(rel|i)p(rated|i)$

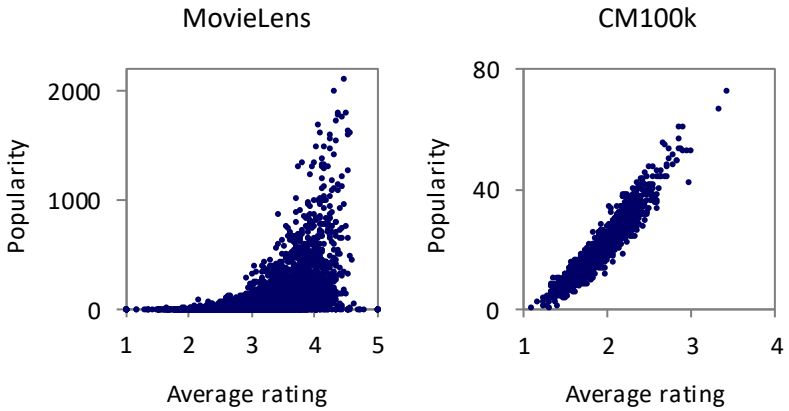

Figure 6.8. Number of relevant ratings (popularity) vs. average rating of each item in MovieLens dataset (left) and CM100k dataset (right).

(case a of Table 4.4). Thus, by removing the bias of the $rated$ distribution we are eliminating one of the two potential sources of divergence between the optimal ranking and the random recommendation. The other, however, is still present: the bias of the relevance distribution. Even though it is not so sharp as the one of MovieLens (see again Figure 4.9 of Chapter 4), it provides enough relevance signal to obtain some advantage from random.

Another important observation to point out is that all the user-based kNN variants are at popularity-level (the item-oriented approaches are even worse). This is what we warned about before, that kNN might be reduced to popularity in absence of discovery and rating decision biases. Note that this cannot be directly inferred from the formal analysis, since some level of dependency between user tastes might be possible even if we remove other biases. In other words, that the specific rating values that each user gives to their randomly assigned songs could have provided enough information to distinguish some users from others. However, that is not the case with CM100k dataset, where such user taste signal is not enough for clearly grouping them, for making kNN producing better recommendations that the non-personalized popularity approaches. Moreover, it seems that matrix factorization is suffering from a similar phenomenon. This makes sense, since matrix factorization is also based on detecting patterns in user ratings, including those derived from the discovery and rating decision processes.

Thus, this experiment may suggest that there is not such a strong dependency between real user preferences as the one we see between the observed ones. Consequently, the best algorithms of the state of the art might not be performing as well as we observe in common offline experiments. However, CM100k is a dataset of small dimensions and more judgments – and perhaps further experiments of similar nature – would be needed to confirm these trends. In any case, what these results certainly verify is that the lower the level of user dependency, the lower the performance of kNN.

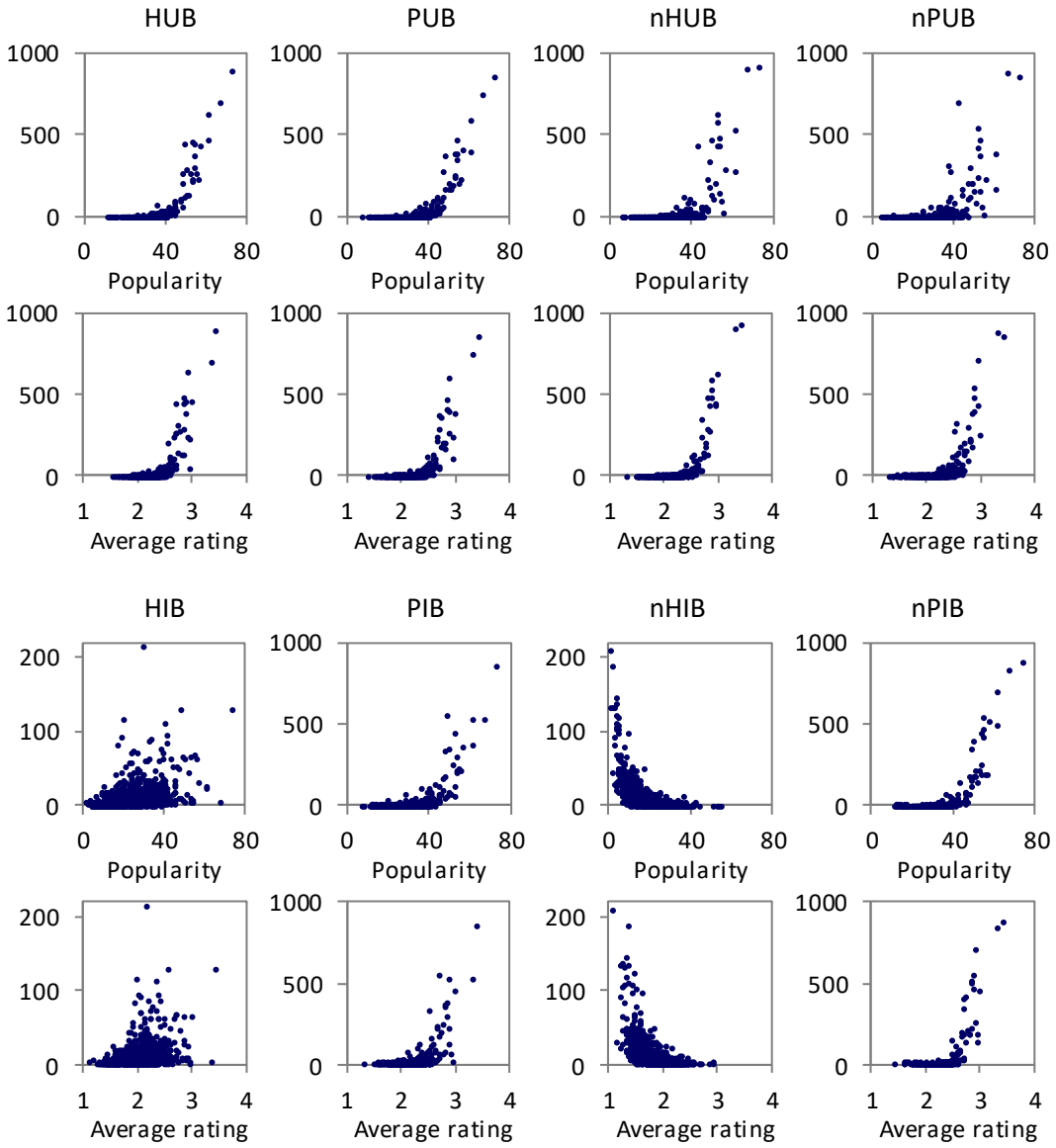

Figure 6.9. Bias towards relevant popularity and average rating the heuristic and probabilistic kNN variants on CM100k. The *x* and *y* axes have the same meaning as in Figure 6.5.

Regarding average rating, we can see that it performs slightly better than relevant popularity (we already checked this in Section 4.8.2 of Chapter 4). Both criteria, however, present quite a similar behaviour in this dataset, as we can verify in Figure 6.8. Such figure depicts for each item its number of relevant ratings versus its average rating, on MovieLens and CM100k. We can see that in CM100k both values are almost proportional, and indeed the item with the largest average rating is also the one with the most relevant ratings, unlike MovieLens where such item has very few ratings.

Following the sign of the comparison between average rating and relevant popularity, and quite remarkably, the user-based normalized kNN variants slightly outperform their

non-normalized counterparts. This atypical behaviour agrees with the analytical trends we set out before, where normalized user-based variants align with average rating whereas non-normalized ones follow relevant popularity. Such trends can also be confirmed in Figure 6.9, where the high correlation between average rating and relevant popularity makes all user-based versions and the probabilistic item-based variants present a strong dependency with both signals. In the heuristic item-based approach (HIB) the bias towards popularity is weaker, but still there to some extent. On the contrary, nHIB is completely opposite to popularity (and thus to average rating), which probably explains its poor performance in Figure 6.7.

To sum up, we have seen a comparative disagreement between results of MovieLens, Last.fm and Netflix, and the ones of CM100k, particularly regarding the comparison of non-normalized vs. normalized kNN variants. Again, the question that arises is whether the low performance of normalized kNN variants on typical datasets is real or a consequence of the experimental design.

6.5 Conclusions

In this chapter, we have developed a fully probabilistic reformulation of the k nearest neighbor approach. A formalization of a recommendation method supposes a significant achievement in itself, since it provides a better explanation of their properties, strengths and potential weaknesses. Moreover, it helps to express the precise hypothesis in which the method is supported, and therefore deduce how it would behave if such hypothesis were not met.

In the case of kNN, the probabilistic development has explicitly revealed its connection with popularity, the initial purpose for which we carried out the formulation. Particularly, we have proved that non-normalized variants align with relevant popularity whereas normalized ones do it with average rating. These different popularity trends suggest a potential explanation of the typical better results obtained by non-normalized kNN variants in standard offline experiments. After all, we have proved in Chapter 4 that popularity-based recommendations are generally rewarded by offline observed metrics, so are expected the algorithms that use the popularity signal to some extent. However, in those situations where average rating results in a truly better option than popularity – as was the case with most of the studied situations of Chapter 4 – we may expect normalized variants performing better that non-normalized ones. In fact, that is the comparison result that we obtain when evaluating with the rating data of CM100k dataset. A revision of common observed results is therefore once again suggested by the analysis and results of this thesis, this time regarding the comparison non-normalized vs. normalized kNN variants.

The formulation has also given rise to other important findings regarding the behaviour of kNN, beyond its popularity biases. Thus, we have proved that its effectiveness

relies on the level of pairwise statistical dependence between user ratings. When such dependency does not exist, kNN is reduced to its corresponding non-personalized trend, relevant popularity or average rating. In this line, using CM100k dataset we have seen an empirical example where, when we remove the artificial patterns caused by discovery and rating decision biases, the remaining relevance dependencies are not enough for kNN to outperform popularity. This experiment suggests a new source of potential disagreement between observed and real results, a new bias that lies once again on the distortion of the observed relevance distribution by discovery and rating decision biases, but focus this time on the generation of artificial – in the sense of alien to user tastes – dependencies between users. However, we consider the study of this new bias and its effects, as future work of this thesis.

Chapter 7

Conclusions and future work

In this thesis we have analysed the role that popularity plays in the recommendation algorithms and evaluation, by studying the behaviour and effectiveness of popularity in its different variants, as well as the potential effects of popularity biases in offline experiments. We have carried out a formal analysis of such effectiveness, expressing it as a function of discovery, relevance and rating distributions, characterizing and studying different situations based on the dependencies between these three variables. We have further delved into the general case without independence assumptions, in the context where item discovery mainly takes place through communication in a social network. This way we studied how social behaviour may determine the shape of the rating final distribution and thus the effectiveness of popularity-based recommendations. Finally, we analyse the popularity component in the k nearest neighbour collaborative filtering scheme, by developing a probabilistic formulation of the kNN approach where the connection to popularity can find a rigorous explanation.

We now present the main conclusion derived from our research; we summarize the work of the thesis and the main contributions arisen from it, and we discuss possible research directions to address in future work.

7.1 Summary and contributions

We summarize and discuss next the main findings and contributions of this thesis, addressing the research goals stated in Chapter 1.

7.1.1 Popularity bias in evaluation

We have formally proved (Chapter 4) that the bias of the popularity distribution (i.e. the distribution of the number of ratings per item) has a direct impact on the observed effectiveness obtained by majority-based recommendation, in particular by relevant popularity. We find that the larger the bias the higher the observed difference between popularity-

based and random recommendation. If this bias is artificially caused by unequal discovery levels of items (for instance, due to different advertising actions), the common methodology in offline experiments may be wrongly rewarding popularity and making it obtain better results than other algorithms, which may be in truth better.

In fact, we have characterized situations in which this contradiction between observed and true metrics actually happens. Such contradictions arise from artificial discovery biases that are not only based on user's tastes, but also depend on other particularities of the items. Unfortunately, these potential contradictions are not restricted to pure popularity-based recommendations. We have illustrated how such contradictions can occur in the comparison of personalized methods, such as two kNN variants.

Therefore, our findings suggest that a revision of the offline evaluation results could be worthwhile. Evaluation with unbiased data can enable fair comparisons between algorithms in such a way that the effect of popularity distributions (whether favourable or counterproductive to recommendation effectiveness) are properly accounted for.

7.1.2 Popularity variants

The formal analysis carried out in Chapter 4 has also allowed us to conclude some general trends for the comparative effectiveness of the different popularity variants. First, we confirmed that relevant popularity is a better option than total popularity for all cases and purposes. It is more robust to user behaviours or discovery biases that run against relevance.

On the other hand, average rating seems to be, as a general trend, a more reliable signal than the number of (relevant) ratings. It presents near-optimal results even in such atypical situations where behaviours and biases work in opposition to users' tastes. Moreover, it shows a better performance in expectation in scenarios with mixed dependencies, where the discovery of the items depends both on the users' tastes and the item itself.

This algorithm has been typically considered as inferior with respect to popularities, since its observed accuracy is clearly worse in common offline experiments. But the recent discovered reward that offline methodologies give to popularity, together with the analytical results that claim average rating is a better candidate in terms of true precision, suggest we must reconsider this perception and all the conclusions deriving from it.

7.1.3 A new optimal ranking principle

Another result arising from the analysis of Chapter 4 is the statement of a new optimal ranking principle: the Low prior Discovery Recall Principle (LDRP). It states that, in order to obtain the maximum true accuracy (for a non-personalized recommendation algorithm), we should sort the items according to the fraction of users that do not known the items yet but that would like them if they were exposed to them. This principle can be

seen as a revised version of the Probability Ranking Principle (PRP) in Information Retrieval (Robertson 1977) considering the particularities of Recommender Systems (and the exclusion of rated items from recommendations in particular).

7.1.4 Popularity as a social process

In Chapter 5 we empirically verify, in a simulation framework, that social communication can play a key role in the formation of popularity distributions and thus, in the performance of recommender systems that are exposed to such distributions. We observed indeed that increasing the communication level between users produces more biased popularity distributions, and therefore an increase of the observed performance of popularity (including, to a lesser extent, the average rating).

Interestingly, the effect is quite the opposite in terms of true effectiveness. The three popularity variants display poor true performance at extreme diffusion levels. The average rating seems the most robust variant against this effect (it is the only one standing above random recommendation in all situations), hinting once again the good properties of this variant. When the communication level is moderate, both the average rating and relevant popularity behave close to the optimal non-personalized recommendation.

We find that viral propagation can be a source of contradictions between observed and true metrics. The average rating achieves a higher true accuracy than relevant popularity in most situations; however, a high diffusion level unfairly rewards the observed performance of relevant popularity. In this scenario, the wrong system could be declared as the winner.

The shape of the social network may also significantly influence how effective or ineffective popular recommendations are. We have shown examples where certain social network structures can boost or slow down the speed and distance at which items reach people in the network, thus shaping the discovery distribution and hence the balance between the variables that determine the performance of popularity in recommendation in one way or the opposite.

7.1.5 Unbiased observation dataset

The construction of the CM100k dataset (http://ir.ii.uam.es/cm100k) is an additional buy-product of this thesis. The dataset presents two special features that distinguish it from other public datasets for recommender system evaluation: first, it does not present any observation bias – discovery, ratings and relevance knowledge are sampled uniformly at random; second, it contains information about the discovery distribution (i.e. what user knows what item). These properties together support both unbiased offline evaluation, and the recreation of typical biased offline experiments, in such a way that the outcome of both methodologies can be compared for the same experiment.

7.1.6 Implications in collaborative filtering algorithms

The development of a probabilistic reformulation of the k nearest neighbour approach reveals that in the absence of dependence between users tastes, kNN is reduced to a popularity-based recommendation. Moreover, depending on the kNN variant, different popularity variants result: non-normalized kNN becomes relevant popularity, whereas normalized kNN degrades to the average rating. It follows that for different degrees of user-user (or item-item) dependence, kNN displays different degrees of similarity to (of bias towards) the corresponding popularity bias (with full independence leading to full equality – in expectation).

The trend towards some popularity variant can therefore imply a trend in kNN towards the properties we have found and proved for pure popularity. For instance, contrary to previous results, we find that normalized kNN can display better accuracy than non-normalized kNN in certain unbiased evaluation conditions (mimicking the corresponding comparative patterns between popularity and the average rating). This would call for a second look at other state of the art collaborative filtering algorithms –and their comparative effectiveness – in a similar – or different – analytical and empirical approach.

Finally, the probabilistic kNN formulation is in itself a by-product contribution of this thesis, since it provides a better explanation of the behaviour of kNN and the hypothesis on which it relies. The formal version can enable further analysis and principled elaborations of the kNN scheme, beyond the work and goals undertaken here.

7.2 Future work

The findings of this thesis open the way for many work lines beyond the results presented here. We describe some of them in the following subsections.

7.2.1 Extension of the formal analysis

The formal analysis developed in Chapter 4 can be extended in many directions. For instance, we analyse the effectiveness of a recommender system in terms of the expectation of $P@1$. Even though we have empirically checked that the conclusions of our analysis generalize to other metrics and cutoffs, an extension which formally models such metrics and cutoffs would provide further confidence in the conclusions, and/or show different particular outcomes. Likewise, we may want to extent the analysis to personalized recommendation algorithms. After all, the initial expression for the expected precision (Equations 4.2 and 4.3) applies to any ranking (personalized or not). Similarly, the current analysis assumes a random partition of the data into training and test sets, but we may consider other standard ways to divide the data, such as temporal splits.

It is also possible, though challenging, to seek formal answers in the social discovery model developed in Chapter 5, using the same random variables, as an alternative or complement to our simulation-based approach.

Further discovery models and elements could be developed and analysed, such as relevance-biased user search, temporal dynamics (e.g. new items and users keep entering the system), discovery feedback loops in recommendations, non-static user preferences (including effects of social influence in preference formation and propagation), etc.

7.2.2 Fair evaluation techniques

The work carried out through this thesis is focused on the analysis and better understanding of the popularity biases and their effects when determining how good a recommender system is. Understanding these biases can be a first step to find better means to cope with them and thus devise more reliable evaluation techniques.

7.2.3 Complex biases

The main object of this thesis is the bias in distributions seen as functions of a single independent variable: the item (its popularity, its discovery, its relevance, etc.). Yet more complex biases can exist – involving, for instance, two or more variables. In particular, in the our probabilistic analysis of kNN we identified a condition involving two variables: a target user (or item) and one of its potential neighbours. The effectiveness of kNN as a personalized collaborative filtering method relies on the pairwise non-independence between user ratings (informally, the "similarity" to potential neighbours should not be uniformly distributed). The same as popularity relies on non-uniform rating and relevance distributions over items, kNN relies on irregularities in the conditional rating and relevance distributions over user pairs. However, the same as the rating distribution over items can be artificially biased, the irregularities in user similarities could be likewise caused by observational biases, unrelated to user preferences.

For instance, if some users have been exposed to the same advertising actions, they might concentrate their ratings on the same products (those promoted by the campaign), thus inflating their similarity in the eyes of the recommendation algorithm. While the real opinion of such users is not necessarily playing a role in the creation of this connection between them.

This would motivate future research on more complex biases, such as observational biases and distorted system perceptions of dual inter-user and inter-item conditional distributions.

7.2.4 User studies

Chapter 4 theorizes on possible situations and patterns in user behaviour, the processes by which they run into experiences, and the influence that user preferences play in these

processes. While our analysis does not make any assumption about what patterns are more or less likely to occur in reality, appropriate user studies could shed light on this point: e.g. to what extent the frequency to discover and provide feedback on a certain experience depends – and in what direction – on whether the experience would be positive or negative, how domain dependent this is, what are common shapes of the discovery and relevance distributions, and so forth. Similarly, a user study to check what trends are observed in practice in specific social environments would be highly helpful in the analysis of social communication effects studied in Chapter 5.

Finally, a dataset sharing the especial characteristics of CM100k but with larger dimensions would be desirable to confirm (or revise) the results we have reported in this thesis. In particular, one with full relevance knowledge where the opinion of every user about every item was available would support the computation of exact true metric values (rather than estimates).

Appendix A

Introducción

A.1 Motivación

Desde sus inicios a principios de los 90, los denominados sistemas de recomendación (Adomavicius & Tuzhilin 2005) han ido progresivamente extendiendo su presencia en las tecnologías de uso diario, hasta suponer hoy en día un elemento familiar para los usuarios de aplicaciones, servicios y herramientas de los ámbitos más cotidianos. La mayor parte de los usuarios estamos acostumbrados hoy a que Youtube nos recomiende videos relacionados con nuestros intereses, a que Spotify nos sugiera nueva música que escuchar, a que Twitter, Linkedin o Facebook nos recomiende contactos con los que conectar, a que Google Play nos sugiera aplicaciones para nuestros smartphones o a que cualquier plataforma de venta online (Amazon, Fnac, etc.) nos recomiende productos que el sistema predice que nos pueden interesar en base a nuestras interacciones previas con la plataforma.

Conceptualmente, un recomendador es un elemento que observa las interacciones del usuario con el sistema y trata de adivinar sus intereses para, posteriormente, sugerirle nuevas opciones que puedan resultarse útiles o interesantes. Implícita en este concepto está la idea de que la satisfacción del usuario será mayor cuanto más personalizadas y adaptadas a sus gustos sean las recomendaciones. La investigación en el campo de los sistemas de recomendación ha asumido, de alguna manera, esta idea como cierta. Sin embargo, una de las preguntas que motivan esta tesis es: ¿Hasta qué punto necesitamos la personalización? ¿Cuánto margen de mejora tenemos entre un enfoque no personalizado y uno personalizado? Podemos intuir que las respuestas a estas preguntas pueden ser complejas o depender de diversos factores. Dicha complejidad es precisamente el objetivo de esta tesis.

Las alternativas no personalizadas más efectivas típicamente consisten en opiniones agregadas de los usuarios que reflejan tendencias mayoritarias – lo que se denomina popularidad (Cañamares y Castells 2018a). Las señales mayoritarias más comunes son el número de personas que ha consumido un producto, que ha mostrado aprecio por él, o el

ratio entre los dos valores anteriores. Las sugerencias no personalizadas basadas en popularidad son de hecho bastante utilizadas hoy en día y podemos encontrarlas prácticamente en cualquier aplicación que involucre un catalogo masivo de opciones. Servicios como Amazon, Youtube, periódicos online, redes sociales, etc., tienen en alguna parte una sección mostrando las opciones más populares – lo más visto, lo más leído, lo más comprado, etc. Más aún, la comunidad investigadora ha descubierto (Cremonesi et al. 2010) que el número de personas a las que satisface una opción popular se encuentra en el mismo orden de magnitud que las que pueden resultar satisfechas por las mejores recomendaciones personalizadas del estado del arte. De hecho, las recomendaciones basadas en popularidad pueden ser la mejor opción posible en aquellas situaciones en las que los datos de interacción disponibles son muy escasos y no proporcionan suficiente información para que los algoritmos personalizados produzcan una recomendación satisfactoria para cada usuario.

Desde una perspectiva más amplia, probar (y por tanto recomendar) aquello que gusta a más gente puede no ser óptimo, pero parece al menos una idea razonable que puede resultar útil en muchos casos. De hecho, la adopción del comportamiento, la opinión o los descubrimientos de otras personas resulta útil para beneficiarnos de la experiencia y el conocimiento aprendido por otros, para guiarnos en situaciones de incertidumbre, o para reducir el coste de elaborar una decisión partiendo desde cero Bandura 1971, Meltzoff & Prinz 2002, Miller & Dollard 1979). Es mucho, por tanto, en lo que nos parecemos y, en muchos casos, lo que es bueno para unos también lo es para otros. En resumen, tenemos mucho en común con la mayoría de nuestros semejantes.

Incluso cuando decidimos personalizar, recientes investigaciones han descubierto que los algoritmos personalizados más efectivos (en particular los métodos de filtrado colaborativo) sesgan fuertemente sus recomendaciones hacia opiniones mayoritarias. La popularidad parece por tanto una tendencia de la que no podemos escapar si pretendemos conseguir recomendaciones efectivas. Peor aún, las metodologías de evaluación offline parecen favorecer la recomendación de opciones populares, lo cual plantea la duda de si los algoritmos están siendo adecuadamente comparados y el estado del arte adecuadamente establecido.

Todos estos comportamientos motivan la investigación llevada a cabo en esta tesis enfocada en avanzar hacia un mejor entendimiento del efecto de la popularidad en el desarrollo, el comportamiento y la evaluación de metodologías de recomendación.

En resumen, la investigación propuesta en esta tesis aborda las siguientes cuestiones:

- ¿Es la popularidad una señal realmente eficaz para producir recomendaciones acertadas?
- ¿Depende la respuesta a la pregunta anterior de la variante de popularidad que se considere? Por ejemplo, ¿habría diferencia entre computar la popularidad de un producto como el número de personas a las que gusta o que lo consumen frente hacerlo como el ratio de consumidores a los que gusta?

- ¿Podemos identificar las condiciones fundamentales de las que depende la respuesta a las presuntas anteriores? Por ejemplo, ¿Depende la efectividad de la popularidad de cómo los usuarios descubren los ítems? ¿O podría depender de su comportamiento, es decir, de si son más propensos a manifestar preferencias positivas que negativas?
- ¿Cómo generaliza esto a los algoritmos de filtrado colaborativo del estado del arte?
- ¿A la hora de comparar dos o más algoritmos, podría llegar a existir una discrepancia entre hacerlo en términos de efectividad medida o en términos de efectividad real? Es decir, ¿Podríamos estar declarando como ganador de la comparativa a un algoritmo que no lo es realmente?

A.2 Objetivos

En base al contexto y las preguntas formuladas anteriormente, el objetivo general de esta tesis consiste en estudiar en qué medida y bajo qué circunstancias la recomendación basada en popularidad es una técnica efectiva o no. Para avanzar hacia dicho objetivo, el trabajo se divide en los siguientes objetivos específicos:

- **O1.** Identificar un conjunto reducido de variables aleatorias que posibilite la descripción de los elementos analizados: distribuciones de popularidad, acierto de la recomendación, descubrimiento de productos, interacciones usuario-ítem, gustos de los usuarios y observaciones del sistema. En base a dichas variables aleatorias, formalizar la distinción entre el acierto real y observado que consigue una recomendación y describir los criterios óptimos no personalizados que maximizan cada uno de estos dos tipos de acierto.
- **O2.** Identificar escenarios prototípicos que pueden ser descritos en términos de las variables aleatorias identificadas y de las dependencias probabilísticas entre ellas, y para los cuales sea posible demostrar la efectividad alcanzada por la recomendación basada en popularidad, en sus distintas variantes.
- **O3.** Verificar los resultados teóricos de forma empírica.
- **O4.** Entender formalmente la influencia de la popularidad en los algoritmos de filtrado colaborativo. Como mencionamos en la motivación, es bien conocido el sesgo que los algoritmos del estado de arte presentan hacia la recomendación de las opciones populares. Sin embargo, las razones de este sesgo no han sido formalmente analizadas ni explicadas todavía.

A.3 Contribuciones

El trabajo desarrollado en esta tesis ha dado lugar a varias contribuciones referentes a la evaluación de los sistemas de recomendación, contribuciones que resumimos a continuación.

- Un marco probabilístico que permite el análisis formal del comportamiento y la efectividad de la recomendación. Este marco modeliza los elementos que determinan la efectividad de diferentes variantes de la popularidad como variables aleatorias explícitas. El comportamiento de la popularidad, y la concordancia de la evaluación offline con la efectividad real, pueden por tanto ser enunciados en términos de dependencias probabilísticas entre dichas variables.
- Como parte del análisis formal, enunciamos un nuevo principio que estable cual es la mejor recomendación no personalizada posible. Adaptamos para ello el principio PRP (Probability Ranking Principle) del campo de la Recuperación de Información (Robertson 1977) a la tarea de recomendación, considerando que los ítems ya conocidos por el usuario no son recomendables. Derivamos dicho principio a partir de una formalización de la efectividad esperada (observada y real) de un recomendador genérico.
- La descripción de situaciones representativas en términos de dependencias probabilísticas entre variables aleatorias. Así como una demostración formal del rendimiento alcanzado en tales situaciones por las distintas variantes de popularidad, comparándolas en términos tanto observados como reales.
- Nuevos hallazgos con respecto a la efectividad relativa de distintas versiones de popularidad. Mientras que el volumen de observaciones (número de votos) suele presentar los mejores resultados en experimentos offline típicos, demostramos que el volumen de interacciones positivas es una señal más fiable y, más importante aún, que el ratio de interacciones positivas (voto promedio) tiende a producir en realidad los mejores resultados cuando se evalúa con datos no sesgados.
- Verificación empírica de la influencia que los fenómenos de difusión de información pueden llegar a tener en la distribución observada de la popularidad y, por tanto, en el comportamiento y el rendimiento de los recomendadores. En particular, vemos que niveles de difusión extremos pueden hacer que la recomendación por popularidad obtenga peores resultados que la recomendación aleatoria.
- Una versión probabilística del algoritmo de vecinos próximos (kNN). La reformulación formal de este algoritmo clásico es útil en sí misma (más allá de su uso en esta tesis) pues permite llevar a cabo todo tipo de análisis y mejoras (suavizados, incorporación de nuevas variables, etc.). En esta tesis nos ha permitido verificar las hipótesis en las que se sustenta kNN, esto es, la dependencia entre los gustos de los usuarios.

- Demostración formal de la conexión entre kNN y la recomendación por popularidad. La reformulación probabilística de kNN que mencionamos anteriormente muestra que, en ausencia de dependencias entre los gustos de los usuarios, kNN se reduce a popularidad. De hecho, distintas variantes de kNN dan lugar a distintas versiones de popularidad.
- Un nuevo conjunto de datos que contiene valoraciones de usuarios reales sobre canciones. Debido a su proceso de recopilación, dicho conjunto nos permite – a nosotros y a la comunidad investigadora – llevar a cabo le evaluación de un algoritmo en ausencia de sesgos externos de popularidad, más allá de aquellos que reflejan los gustos reales de los usuarios. Más aún, el conjunto contiene información acerca de la distribución de descubrimiento, lo que a su vez permite la recreación de un experimento estándar de evaluación offline donde además están disponibles juicios de relevancia extra para computar la efectividad real y compararla con la observada.

La mayor parte de las contribuciones anteriores se enfocan en el estudio de los sesgos de popularidad y sus efectos. El entendimiento de dichos sesgos es un primer paso para encontrar mejores formas de lidiar con ellos y, por tanto, desarrollar técnicas de evaluación más fiables (Castells y Cañamares 2018), una de las principales líneas de trabajo futuro de esta tesis.

A.4 Estructura de la tesis

La tesis se estructura de la siguiente manera:

- El Capítulo 1 (Introducción) presenta la motivación, los objetivos, las contribuciones y las publicaciones relacionadas con esta tesis.
- El Capítulo 2 (Observaciones preliminares) presenta una revisión preliminar y empírica de la efectividad de la popularidad. En primer lugar, explicamos la tarea de recomendación e introducimos el concepto de popularidad y las posibles interpretaciones que tiene. A continuación, comparamos su efectividad con la de otros algoritmos representativos del estado del arte y, por último, verificamos la existencia en dichos algoritmos de sesgos hacia los productos populares.
- El Capítulo 3 (Trabajo relacionado) expone y analiza los trabajos y estudios anteriores que están relacionados con la popularidad. Introducimos una serie de conceptos relacionados con los sistemas de recomendación y sus evaluación, y agrupamos los estudios previos en función del aspecto de la popularidad que abordan.
- El Capítulo 4 (Sesgos de popularidad en la evaluación de los sistemas de recomendación) lleva a cabo un análisis formal de la efectividad alcanzada en distintas situaciones por las diferentes interpretaciones de la popularidad. Proponemos un marco probabilístico sobre el cual seguidamente expresamos la efectividad esperada (observada

y real) de un algoritmo en términos de una serie de factores que nos permiten caracterizar distintas situaciones. Aplicando la expresión anterior al caso particular de la recomendación por popularidad estudiamos los factores de los que su efectividad depende. Además, también empleamos la expresión de la efectividad esperada de un recomendador para deducir el criterio óptimo no personalizado que maximiza dicha efectividad.

- El Capítulo 5 (Sesgos de popularidad derivados de fenómenos de red social) profundiza en una de las situaciones analizadas en el Capítulo 4 mediante la simulación del descubrimiento y la interacción con los ítems a través de la comunicación en una red social. Observamos cómo diferentes aspectos de la interacción social pueden afectar a las observaciones que se encuentran disponibles (como entrada) de los recomendadores y, por tanto, a su efectividad.
- El Capítulo 6 (Sesgos de popularidad en el algoritmo de vecinos próximos) estudia la conexión entre el algoritmo de vecinos próximos (kNN) y la recomendación por popularidad. Para ello, proponemos una reformulación probabilística de este método de filtrado colaborativo que, además de mostrar explícitamente su conexión con la popularidad, nos permite expresar las hipótesis principales en las que se sustenta kNN.
- El Capítulo 7 (Conclusiones y trabajo futuro) resume el trabajo expuesto en la tesis y sintetiza las conclusiones que se derivan de él. Introducimos además las posibles líneas a seguir como trabajo futuro.
- El Apéndice A contiene la traducción al español del Capítulo 1.
- El Apéndice B contiene la traducción al español del Capítulo 7.

A.5 Publicaciones

El trabajo desarrollado a lo largo de esta tesis ha dado lugar a varias publicaciones en congresos internacionales del área de la Recuperación de Información. A continuación se listan dichas publicaciones, agrupándolas de acuerdo al capítulo de la tesis con el que están relacionadas:

Publicaciones relacionadas con el Capítulo 4

Las siguientes tres publicaciones están relacionadas con el análisis formal de la efectividad de la popularidad que desarrollamos en el Capítulo 4.

- R. Cañamares and P. Castells. Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). Ann Arbor, Michigan, USA, July 2018, pp. 415-424.
  CORE A+ (full paper)

Premio a la mejor publicación del congreso.

- R. Cañamares and P. Castells. From the PRP to the Low Prior Discovery Recall Principle for Recommender Systems. 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). Ann Arbor, Michigan, USA. July 2018, pp. 1081-1084.
  CORE A+ (short paper)
- R. Cañamares and P. Castells. On the Optimal Non-Personalized Recommendation: From the PRP to the Discovery False Negative Principle. Workshop on Axiomatic Thinking for Information Retrieval and Related Tasks (ATIR 2017) at the 40th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). Tokyo, Japan, August 2017.

Publicaciones relacionadas con el Capítulo 5

La siguiente publicación aborda la generación de los sesgos de popularidad en el entorno de una red social, así como la forma en que dichos sesgos pueden afectar la efectividad de la recomendación por popularidad.

- R. Cañamares and P. Castells. Exploring social network effects on popularity biases in recommender systems. 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014) at the 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, USA, October 2014.
  CORE A+ (full paper)

Publicaciones relacionadas con el Capítulo 6

En la siguiente publicación de SIGIR 2017 proponemos una reformulación probabilística del algoritmo de vecinos próximos (kNN). Una reformulación que explícitamente expresa la conexión de dicho algoritmo con la popularidad.

- R. Cañamares and P. Castells. A Probabilistic Reformulation of Memory-Based Collaborative Filtering – Implications on Popularity Biases. 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). Tokyo, Japan, August 2017, pp. 215-224.

Otras publicaciones relacionadas con la tesis

De acuerdo con la idea de evitar los sesgos de popularidad, así como cualquier otro tipo de distorsión que induzca a error en las conclusiones de un experimento de evaluación offline, en la siguiente publicación proponemos una metodología experimental para obtener resultados no sesgados.

- P. Castells and R. Cañamares. Characterization of Fair Experiments for Recommender System Evaluation – A Formal Analysis. Workshop on Offline Evaluation for Recommender Systems (REVEAL 2018) at the 12th ACM Conference on Recommender Systems (RecSys 2018). Vancouver, Canada, October 2018.

Appendix B

Conclusiones y trabajo futuro

En esta tesis hemos analizado el papel que la popularidad juega en el funcionamiento y la evaluación de los algoritmos de recomendación, estudiando el comportamiento y la efectividad que presenta en sus diferentes versiones, así como los posibles efectos de los sesgos de popularidad en los experimentos de evaluación offline. Hemos llevado a cabo un análisis formal de dicha efectividad, expresándola como función de las distribuciones de descubrimiento, relevancia y votos, y caracterizando y estudiando distintas situaciones en base a las dependencias entre esas tres distribuciones. Además, hemos profundizado en el caso general, en el que no hay asunciones de independencia, en un contexto donde el descubrimiento de los ítems tiene lugar principalmente a través de la comunicación en una red social. De esta forma hemos estudiado como el comportamiento social puede influir en la forma final de la distribución de votos y, con ello, en la efectividad de las recomendaciones basadas en popularidad. Finalmente, hemos analizado el componente de popularidad en el algoritmo de filtrado colaborativo kNN, desarrollando una formulación probabilística de dicho algoritmo que permite explicar rigurosamente su conexión con la popularidad.

Presentamos ahora las principales conclusiones de nuestra investigación. En primer lugar, resumimos el trabajo desarrollado y las principales contribuciones del mismo, y a continuación proponemos las posibles líneas de investigación a abordar en el futuro.

B.1 Resumen y contribuciones

Resumimos y discutimos a continuación los principales hallazgos y contribuciones de esta tesis, abordando los objetivos establecidos en el Capítulo 1.

B.1.1 Sesgos de popularidad en evaluación

Hemos probado formalmente (Capítulo 4) que el sesgo de la distribución de popularidad (la distribución del número de votos por ítem) tiene un impacto directo en la efectividad observada de la recomendación por mayorías, en particular de la popularidad relevante.

Así, hemos visto que cuanto mayor es el sesgo, mayor es la diferencia observada entre recomendar por popularidad y hacerlo de forma aleatoria. Si dicho sesgo es causado artificialmente por un nivel desigual de descubrimiento de los ítems (por ejemplo, debido a distintas campañas publicitarias), entonces la metodología seguida en los experimentos estándar de evaluación offline podría llegar a favorecer injustamente la efectividad de la popularidad, haciendo que aparentemente obtenga mejores resultados que otros algoritmos que son en realidad mejores.

De hecho, hemos caracterizados situaciones en las cuales dicha contradicción entre los valores observados y reales de las métricas ocurre. Tales contradicciones están relacionadas con sesgos artificiales de descubrimiento que no se basan únicamente en los gustos de los usuarios, sino que dependen de otras particularidades de los ítems. Desgraciadamente, estas contradicciones no se restringen a las recomendaciones por popularidad pura y hemos visto que también pueden ocurrir en la comparación de otros algoritmos personalizados, en concreto entre dos variantes de kNN.

Nuestros hallazgos sugieren, por tanto, que podría interesar llevar a cabo una revisión de los resultados obtenidos mediante metodologías de evaluación offline. La evaluación con datos no sesgados puede permitir comparaciones justas entre algoritmos de tal forma que el efecto (ya sea favorable o perjudicial) de las distribuciones de popularidad en la efectividad de la recomendación sea adecuadamente tenido en cuenta.

B.1.2 Versiones de la popularidad

El análisis formal llevado a cabo en el Capítulo 4 también nos ha permitido concluir algunas tendencias generales acerca de la efectividad relativa de las diferentes variantes de la popularidad. En primer lugar, hemos confirmado que la popularidad relevante es una opción mucho más adecuada que la popularidad total en todas las situaciones, ya que resulta más robusta frente a comportamientos de usuario y sesgos de descubrimiento contrarios a la relevancia.

Por otro lado, el voto promedio parece ser en general una señal mucho más fiable que el número de votos (relevantes o no). Esta opción presenta resultados casi óptimos incluso en situaciones atípicas con comportamientos y sesgos contrarios a los gustos de los usuarios. Más aún, en esperanza muestra un mejor rendimiento que las popularidades en escenarios de dependencias mixtas, donde el descubrimiento de los ítems depende tanto de los gustos de los usuarios como del ítem en sí mismo.

Este algoritmo ha sido considerado inferior a las popularidades, ya que su efectividad observada es claramente peor en experimentos offline. Pero la injusta recompensa que las metodologías de evaluación offline otorgan a la popularidad, y que hemos demostrado formalmente en esta tesis, junto con los resultados analíticos que afirmar que el voto promedio es un mejor candidato en términos de efectividad real, sugieren que debemos reconsiderar esta percepción y todas las conclusiones a las que haya dado lugar.

B.1.3 Un nuevo principio de ranking óptimo

Otro resultado que se deriva del análisis llevado a cabo en el Capítulo 4 es el enunciado de un nuevo principio de ranking óptimo: el Low prior Discovery Recall Principle (LDRP). Dicho principio establece que, de cara a obtener la máxima precisión real (para un recomendador no personalizado), debemos ordenar los ítems de acuerdo a la fracción de usuarios que no los conocen todavía pero que los considerarían relevantes si lo hicieran. Este principio puede verse como una adaptación del principio PRP (Probability Ranking Principle) del campo de la Recuperación de Información (Robertson 1977), a las particularidades de los sistemas de recomendación (en particular a la exclusión de los ítems ya votados de las recomendaciones).

B.1.4 La popularidad como proceso social

En el Capítulo 5 verificamos empíricamente mediante simulaciones que la comunicación social puede jugar un papel fundamental en la formación de las distribuciones de popularidad y, con ello, en el rendimiento de los recomendadores que están expuestos a dichas distribuciones. Observamos, de hecho, que incrementar el nivel de comunicación entre los usuarios produce distribuciones de popularidad cada vez más sesgadas y, con ello, un incremento en el rendimiento observado de las popularidades (incluyendo el del voto promedio, aunque en menor medida).

Curiosamente, el efecto es justo el opuesto en términos de efectividad real. Así, las tres variantes de la popularidad presentan un rendimiento bastante bajo con niveles de difusión extremos. El voto promedio parece ser el más robusto ante este comportamiento (es el único que obtiene siempre mayor precisión que la recomendación aleatoria), insinuando una vez más las buenas propiedades de esta variante. Cuando el nivel de comunicación es moderado, tanto el voto promedio como la popularidad relevante presentan un comportamiento cercano a la recomendación óptima no personalizada.

Encontramos, por tanto, que un nivel viral de propagación puede llegar a ser una fuente de contradicciones entre valores observadores y reales en las métricas. Así, el voto promedio alcanza una efectividad real superior a la de la popularidad relevante en la mayor parte de las situaciones analizadas, sin embargo, los niveles altos de difusión benefician injustamente el rendimiento observado de esta última, haciendo que obtenga una aparente mejor efectividad. En un escenario como el descrito, estaríamos declarando al candidato incorrecto como ganador de la comparativa.

La estructura de la red social puede también influir significativamente en la efectividad de las recomendaciones por popularidad. Hemos mostrado ejemplos donde ciertas estructuras de red social pueden acelerar o ralentizar la velocidad a la que los ítems llegan a las personas y, con ello, la forma de la distribución de descubrimiento y el equilibro entre las variables que determinan el rendimiento de la popularidad como recomendación.

B.1.5 Conjunto de datos sin sesgo de observación

La recopilación del conjunto de datos CM100k (http://ir.ii.uam.es/cm100k) es otra contribución importante de esta tesis. Este conjunto presenta dos características especiales que lo distinguen de otros conjuntos de datos públicos disponibles para la evaluación de sistemas de recomendación: en primer lugar, no presenta sesgos de observación – la información de descubrimiento y relevancia es muestreada de forma uniforme y aleatoria – y, en segundo, contiene información acerca de la distribución de descubrimiento, esto es, acerca de qué usuarios conocen qué ítems. Ambas propiedades juntas permiten, tanto realizar una evaluación no sesgada, como recrear un experimento típico de evaluación offline con datos segados, de tal manera que es posible comparar las salidas de ambas metodologías para un mismo experimento.

B.1.6 Implicaciones en los algoritmos de filtrado colaborativo

El desarrollo de una reformulación probabilística del algoritmo de vecinos próximos (kNN) revela que, en ausencia de dependencias entre los gustos de los usuarios, kNN se reduce a la recomendación por popularidad. Más aún, dependiendo de la variante de kNN, éste puede resultar en una variante u otra de popularidad: así, los kNN no normalizados se reducen a la popularidad relevante, mientras que los normalizados se degradan al voto promedio. Se sigue, por tanto, que para diferentes niveles de dependencia usuario a usuario (o ítem a ítem), kNN presenta distintos niveles de similitud con su correspondiente variante de popularidad, obteniéndose la igualdad – en esperanza – con una independencia completa.

La tendencia hacia alguna variante de popularidad puede implicar, por tanto, una tendencia en kNN hacia los comportamientos que hemos descubierto y probado para la popularidad pura. Por ejemplo, hemos mostrado que, contrariamente a lo que se ha venido observando en la literatura, el kNN normalizado puede presentar mayor efectividad que el no-normalizado bajo ciertas condiciones de evaluación no sesgada (imitando la correspondiente comparativa entre el voto promedio y la popularidad relevante). Esto sugeriría revisar otros algoritmos de filtrado colaborativo del estado de arte, y en particular su efectividad relativa, bajo un enfoque analítico y empírico similar.

Finalmente, la reformulación probabilística es en sí misma otra importante contribución de esta tesis, pues supone una mejor explicación del comportamiento de kNN y de las hipótesis en las que se sustenta. La versión formal abre además la puerta a posteriores análisis y elaboraciones fundamentadas del esquema de kNN, más allá de los desarrollados aquí de acuerdo a los objetivos de esta tesis.

B.2 Trabajo futuro

Las contribuciones de esta tesis sientan la base para el desarrollo de muchas líneas de trabajo futuro, algunas de las cuales describimos a continuación.

B.2.1 Extensión del análisis formal

El análisis formal desarrollado en el Capítulo 4 puede ser extendido en muchas direcciones. Por ejemplo, expresamos la efectividad de un recomendador en términos de la esperanza de $P@1$. A pesar de que hemos verificado empíricamente que las conclusiones de dicho análisis generalizan a otras métricas y a rankings de mayor profundidad, una extensión del estudio que incluya la formalización de dichas métricas y profundidades proporcionaría mayor rigor y confianza en las conclusiones, y/o mostraría diferentes resultados particulares. De igual forma, sería deseable extender el análisis a recomendadores personalizados, después de todo, la expresión inicial para la precisión esperada (Ecuaciones 4.2 y 4.3) puede ser aplicada a cualquier ranking, personalizado o no. Por otro lado, se podrían considerar otras posibles formas de dividir los datos en entrenamiento y test (más allá de la partición aleatoria), como puede ser una partición temporal.

También es posible, aunque complejo, buscar respuestas formales en el modelo de descubrimiento social desarrollado en el Capítulo 5, como alternativa o complemento del enfoque basado en simulaciones que hemos seguido en dicho capitulo.

Por otro lado, otros modelos de descubrimiento podrían ser desarrollados y analizados, tales como búsquedas sesgadas por los gustos del usuario, las dinámicas temporales (p.e. ítems y usuarios que entran al sistema en distintos momentos), el efecto retroactivo de los recomendadores, los gustos no estáticos (incluyendo efectos de influencia social en la formación y propagación de opiniones), etc.

B.2.2 Técnicas no sesgadas de evaluación

El trabajo desarrollado en esta tesis se centra en el análisis de los sesgos de popularidad y de sus efectos a la hora de determinar cómo de bueno es un sistema de recomendación. Entender dichos sesgos es un primer paso para lidiar con ellos y desarrollar mejores y más fiables técnicas de evaluación.

B.2.3 Estudio de sesgos más complejos

El principal objetivo de esta tesis es el sesgo de las distribuciones, vistas como funciones de una única variable independiente: el ítem (su popularidad, su descubrimiento, su relevancia, etc.). Pero pueden existir sesgos más complejos que afecten, por ejemplo, a dos o más variables. En particular, en nuestro análisis probabilístico de kNN identificamos una condición que hace referencia a dos variables: el usuario (o ítem) objetivo y uno de sus

posibles vecinos. La efectividad de kNN como algoritmo personalizado de filtrado colaborativo se sustenta en la dependencia entre los votos de los usuarios (informalmente, en que la similitud de los posibles vecinos no se distribuya uniformemente). De igual forma que la popularidad se sustenta en distribuciones de voto y de relevancia no uniformes, kNN se basa en las irregularidades de las distribuciones condicionales de voto y relevancia sobre los pares usuario-usuario. Sin embargo, al igual que el sesgo de la distribución de voto puede ser artificial, las irregularidades en las similitudes entre usuarios también puedes estar causadas por sesgos de observación que no tengan relación con las preferencias de los usuarios.

Por ejemplo, si ciertos usuarios han sido expuestos a las mismas campañas de publicidad, concentraran sus votos en los mismos productos (aquellos promocionados por dichas campañas), lo que incrementará su similitud a ojos del sistema de recomendación. Mientras que las opiniones reales de dichos usuarios no están jugando necesariamente ningún papel en la creación de esta conexión entre ellos.

Esto motivaría la investigación de sesgos más complejos, tales como los sesgos de observación y las percepciones distorsionadas del sistema en las distribuciones condicionales usuario-usuario o ítem-ítem.

B.2.4 Estudios con usuarios reales

El Capítulo 4 teoriza acerca de posibles situaciones y patrones en los comportamientos de los usuarios, los procesos a través de los cuales llegan a interactuar con el sistema y la influencia que las preferencias de los usuarios tienen en dichos procesos. Mientras que nuestro análisis no realiza ninguna asunción acerca de qué patrones se dan con mayor o menor frecuencia en la realidad, la realización de estudios con usuarios reales podría aportar información al respecto: por ejemplo, hasta qué punto depende la frecuencia a la que se descubren y votan ciertas experiencias de si dichas experiencias son positivas o negativas, o en qué sentido es dicha dependencia y si depende del dominio, o cuales son las formas más comunes de las distribuciones de descubrimiento y relevancia, etc. Análogamente, un estudio con usuarios reales para comprobar que comportamientos se producen en la práctica en redes sociales específicas sería de gran ayuda en el análisis llevado a cabo en el Capítulo 5 acerca de los efectos de la comunicación social.

Finalmente, sería conveniente disponer de un conjunto de datos de las mismas características que CM100k, pero de mayores dimensiones, para confirmar (o revisar) los resultados obtenidos a lo largo de esta tesis. En particular, un conjunto de datos que contuviera información de relevancia completa, es decir, donde la opinión de cada usuario sobre cada ítem estuviera disponible para calcular el valor exacto de las métricas verdaderas, sin necesidad de aproximación ninguna.

References

A. Abeliuk, G. Berbeglia, P. Van Hentenryck, T. Hogg and K. Lerman (2017). Taming the unpredictability of cultural markets with social influence. In Proceedings of the 26th International Conference on World Wide Web (WWW 2017). Perth, Australia, April 2017, pp. 745-754.

H. Abdollahpouri, R. Burke and B. Mobasher (2017). Recommender Systems as Multistakeholder Environments. In Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization (UMAP 2017). Bratislava, Slovakia, July 2017, pp. 347-348.

P. Adamopoulos and A. Tuzhilin (2014). On unexpectedness in recommender systems: or how to better expect the unexpected. ACM Transactions on Intelligent Systems and Technology 5(4). ACM, New York, NY, January 2015, pp. 1-32.

P. Adamopoulos, A. Tuzhilin and P. Mountanos (2015). Measuring the Concentration Reinforcement Bias of Recommender Systems. In Proceedings of the 9th ACM Conference on Recommender Systems (RecSys 2015), Poster Session. Vienna, Austria, September 2015.

G. Adomavicius and Y. Kwon (2012). Improving aggregate recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering 24(5). IEEE, Piscataway, NJ, USA, May 2012, pp. 896-911.

G. Adomavicius and A. Tuzhilin (2005). Toward the next generation of recommender systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering 17(6). IEEE, Piscataway, NJ, USA, June 2005, pp. 734-749.

F. Aiolli (2013). Efficient top-n recommendation for very large scale binary rated datasets. In Proceedings of the 7th ACM Conference on Recommender Systems (RecSys 2013). Hong Kong, October 2013, pp. 273-280.

R. Baeza and B. Ribeiro (2011). Modern Information Retrieval: The Concepts and Technology behind Search, 2nd Edition. ACM Press Books, 2011.

E. Bakshy, I. Rosenn, C. Marlow and L. Adamic. The Role of Social Networks in Information Diffusion. In Proceedings of the 21st International Conference on World Wide Web (WWW 2012). Lyon, France, April 2012, pp. 519-528.

A. Bandura (1971). Social Learning Theory. General Learning Press, New York, NY, USA, 1971.

A. L. Barabási and R. Albert (1999). Emergence of scaling in random networks. Science 286(5439), October 1999, pp. 509-512.

A. R. Benson, R. Kumar and A. Tomkins (2016). Modeling User Consumption Sequences. In Proceedings of the 25th International Conference on World Wide Web (WWW 2016). Montréal, Canada, April 2016, pp. 519-529.

N. J. Belkin and W. B. Croft. Information filtering and information retrieval: two sides of the same coin? Communications of the ACM - Special issue on information filtering 35(12). ACM, New York, NY, December 1992, pp. 29-38.

A. Bellogín, P. Castells and I. Cantador (2011). Precision-Based Evaluation of Recommender Systems: An Algorithmic Comparison. In Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011). Chicago, Illinois, October 2011, pp. 333-336.

A. Bellogín, P. Castells and I. Cantador (2017). Statistical Biases in Information Retrieval Metrics for Recommender Systems. Information Retrieval 20(6). Springer, Dordrecht, Netherlands, July 2017, pp. 606-634.

S. Bikhchandani, D. Hirshleifer, and I. Welch (1992). A Theory of Fads, Custom, and Cultural Change as Informational Cascades. The Journal of Political Economy 100(5). University of Chicago Press, Chicago, IL, USA, October 1992, pp. 992-1026.

M. Blattner and M. Medo. Recommendation Systems in the Scope of Opinion Formation: a Model. In Proceedings of the 2nd Workshop on Human Decision Making in Recommender Systems in conjunction with the 6th ACM Conference on Recommender Systems (RecSys 2012). Dublin, Ireland, September 2012, pp. 32-39.

Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager and A. Mahanti (2011). Characterizing and Modeling Popularity of User-generated Videos. 29th IFIP WG 7.3 International Symposium on Computer Performance, Modeling, Measurements and Evaluation 2011 (IFIP Performance 2011) 68(11). Elsevier Science Publishers B. V., Amsterdam, Netherlands, October 2011, pp. 1037-1055.

R. Bredereck and E. Elkind (2017). Manipulating Opinion Diffusion in Social Networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI 2017). Morgan Kaufmann Publishers, San Francisco, CA, USA, pp. 894-900.

J. Bryant and M. B. Oliver (Eds.) (2008). Media Effects: Advances in Theory and Research, 3rd edition. Routledge, Abingdon, UK, 2008.

S. Büttcher, C. L. A. Clarke and G. V. Cormack (2010). Information Retrieval: Implementing and Evaluating Search Engines. The MIT Press, Cambridge, Massachusetts, USA, 2010.

R. Cañamares and P. Castells (2018a). Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems. In Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). Ann Arbor, Michigan, USA, July 2018, pp. 415-424.

R. Cañamares and P. Castells (2018b). From the PRP to the Low Prior Discovery Recall Principle for Recommender Systems. In Proceedings of the 41st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018). Ann Arbor, Michigan, USA. July 2018, pp. 1081-1084.

R. Cañamares and P. Castells (2017a). A Probabilistic Reformulation of Memory-Based Collaborative Filtering – Implications on Popularity Biases. In Proceedings of the 40th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). Tokyo, Japan, August 2017, pp. 215-224.

R. Cañamares and P. Castells (2017b). On the Optimal Non-Personalized Recommendation: From the PRP to the Discovery False Negative Principle. Workshop on Axiomatic Thinking for Information Retrieval and Related Tasks (ATIR 2017) at the 40th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2017). Tokyo, Japan, August 2017.

R. Cañamares and P. Castells (2014). Exploring social network effects on popularity biases in recommender systems. 6th Workshop on Recommender Systems and the Social Web (RSWeb 2014) at the 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, USA, October 2014.

P. Castells and R. Cañamares (2018). Characterization of Fair Experiments for Recommender System Evaluation – A Formal Analysis. Workshop on Offline Evaluation for Recommender Systems (REVEAL 2018) at the 12th ACM Conference on Recommender Systems (RecSys 2018). Vancouver, Canada, October 2018.

P. Castells, N. J. Hurley, S. Vargas (2015). Novelty and Diversity in Recommender Systems. In Recommender Systems Handbook, 2nd edition, F. Ricci, L. Rokach, and B. Shapira (Eds.). Springer, 2015, pp. 881-918.

O. Celma (2010). Music Recommendation. In Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space. Springer-Verlag Berlin Heidelberg, 2010, pp.43-85.

O. Celma and P. Herrera (2008). A new approach to evaluating novel recommendations. In Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys 2008). Lousanne, Switzerland, October 2008, pp. 179-186.

Z. Cheng and N. Hurley (2009). Effective diverse and obfuscated attacks on model-based recommender systems. In Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys 2009). New York, NY, USA, October 2009, pp 141-148.

R. B. Cialdini and N. J. Goldstein (2004). Social Influence: Compliance and Conformity. Annual Review of Psychology 55. Palo Alto, CA, USA, February 2004, pp. 591-621

G. L. Ciampaglia, A. Nematzadeh, F. Menczer and A. Flammini (2018). How algorithmic popularity bias hinders or promotes quality. Scientific Reports 8(1). Nature Research, London, UK, October 2018.

P. Cremonesi, Y. Koren and R. Turrin (2010). Performance of recommender algorithms on top-n recommendation tasks. In Proceedings of the 4th ACM Conference on Recommender Systems (RecSys 2010). Barcelona, Spain, September 2010, pp. 39-46.

P. Cremonesi, F. Garzotto, R. Pagano and M. Quadrana (2014). Recommending without short head. In Proceedings of the 23rd International Conference on World Wide Web (WWW 2014 Companion Volume). Seoul, Republic of Korea, April 2014, pp. 245-246.

B. Doerr, M. Fouz and T. Friedrich (2012). Why rumors spread so quickly in social networks. Communications of the ACM 55(6). ACM, New York, NY, Jan 2012, pp. 70-75.

C. Dwork, M. Hardt, T. Pitassi, O. Reingold and R. S. Zemel (2012). Fairness through awareness. In Proceedings of the 3rd Innovations on Theoretical Computer Science Conference (ITCS 2012). Bangalore, India, January 2012, pp. 214-226.

D. Fleder and K. Hosanagar (2009). Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity. Management Science 55(5). Informs, Catonsville, MD, USA, May 2009, pp. 697-712.

A. Gilotte, C. Calauzènes, T. Nedelec, A. Abraham and S. Dollé (2018). Offline A/B Testing for Recommender Systems. In Proceedigs of the 11th ACM International Conference on Web Search and Data Mining (WSDM 2018). Los Angeles, California, USA, February 2018, pp. 198-206.

S. Goel, A. Broder, E. Gabrilovich and B. Pang (2010). Anatomy of the long tail: ordinary people with extraordinary tastes. In Proceedigs of the 3th ACM International Conference on Web Search and Data Mining (WSDM 2010). New York, NY, USA, February 2010, pp. 201-210.

A. Gruson, P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu and Ben Carterette (2019). Offline Evaluation to Make Decisions About Playlist Recommendation. In Proceedings of the 12th ACM International Conference on Web Search and Data Mining (WSDM 2019). Melbourne, VIC, Australia, February 2019, pp. 420-428.

F. M. Harper and J. A. Konstan (2016). The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5(4). ACM, New York, NY, USA, January 2016.

F. M. Harper, X. Li, Y. Chen and J. A. Konstan (2005). An Economic Model of User Rating in an Online Recommender System. In Proceedings of the 10th International Conference on User Modeling (UM 2005). Edinburgh, Scotland, UK, July 2005, pp. 307-316.

H. He and E. A. Garcia (2009). Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering 21(9). IEEE, Piscataway, NJ, USA, September 2009, pp. 1263-1284.

E. Hensinger, I. Flaounas and N. Cristianini (2013). Modelling and predicting news popularity. Pattern Analysis & Applications 16(4). Springer, Dordrecht, Netherlands, November 2013, pp. 623-635.

J. L. Herlocker, J. A. Konstan, L. G. Terveen and J. T. Riedl (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems (TOIS) 22(1). ACM, New York, NY, USA, January 2004, pp. 5-53.

Y. Hu, Y. Koren and C. Volinsky (2008). Collaborative filtering for implicit feedback datasets. In Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008). Pisa, Italy, December 2008, pp. 263-272.

D. Jannach, L. Lerche, I. Kamehkhosh, and M. Jugovac (2015). What recommenders recommend: an analysis of recommendation biases and possible countermeasures. User Modeling and User-Adapted Interaction 25(5). Springer, Dordrecht, Netherlands, December 2015, pp. 427-491.

J. Kawale, H. H. Bui, B. Kveton, L. Tran-Thanh and S. Chawla (2015). Efficient Thompson Sampling for Online Matrix-Factorization Recommendation. In Proceedings of Neural Information Processing Systems (NIPS 2015). Curran Associates, Inc., Red Hook, NY, USA.

S. Krishnan, J. Patel, M.J. Franklin and K. Goldberg (2014). Social Influence Bias in Recommender Systems: A Methodology for Learning, Analyzing, and Mitigating Bias in Ratings. In Proceedings of the 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, Silicon Valley, USA, October 2014, pp. 137-144.

K. Lee and K. Lee (2011). My head is your tail: applying link analysis on longtailed music listening behavior for music recommendation. In Proceedings of the 5th ACM Conference on Recommender Systems (RecSys 2011). Chicago, Illinois, October 2011, pp. 213-220.

S. Li, A. Karatzoglou, and C. Gentile (2016). Collaborative Filtering Bandits. In Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2016). Pisa, Italy, July 2016, pp. 539-548

G. Linden, B. Smith and J. York (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. IEEE Internet Computing 7(1). IEEE, Piscataway, NJ, USA, January 2003, pp. 76-80.

R. J. A. Little and D. B. Rubin (1987). Statistical analysis with missing data. John Wiley & Sons, Hoboken, NJ, USA, 1987.

B. Marlin and R. Zemel (2009). Collaborative prediction and ranking with non-random missing data. In Proceedings of the 3rd ACM Conference on Recommender Systems (RecSys 2009). New York City, NY, USA, October 2009, pp. 5-12.

B. Marlin, R. Zemel, S. Roweis, and M. Slaney (2007). Collaborative filtering and the missing at random assumption. In Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence (UAI 2007). Vancouver, BC Canada, July 2007, pp. 267-75.

J. J. McAuley and J. Leskovec (2012). Learning to Discover Social Circles in Ego Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems (NIPS 2012). Lake Tahoe, NV, USA, December 2012, pp. 548-556.

S. M. McNee, J. Riedl and J. A. Konstan (2006). Being accurate is not enough: how accuracy metrics have hurt recommender systems. In Extended Abstracts on Human Factors in Computing Systems (CHI EA 2006). Montréal, Québec, Canada, April 2006, pp. 1097-1101.

R. Mehrotra, J. McInerney, H. Bouchard, M. Lalmas and F. Diaz (2018). Towards a Fair Marketplace: Counterfactual Evaluation of the Trade-off between Relevance, Fairness

and Satisfaction in Recommendation Systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (CIKM 2018). Lingotto, Turin, Italy, October 2018, pp. 2243-2251.

A. N. Meltzoff and W. Prinz (2002). The imitative mind: Development, evolution, and brain bases, 1st Edition. Cambridge University Press, 2002.

N. E. Miller and J. Dollard (1979). Social Learning and Imitation, New edition. Greenwood Press Reprint, 1979.

M. Moussaïd, J. E. Kämmer, P. P. Analytis and H. Neth (2013). Social Influence and the Collective Dynamics of Opinion Formation. PLOS ONE 8(11). Public Library of Science, San Francisco, CA, USA, November 2013.

S. A. Myers, C. Zhu, and J. Leskovec. Information Diffusion and External Influence in Networks. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2012). Beijing, China, August 2012, pp. 33-41.

M. Nakatsuji, Y. Fujiwara and A. Tanaka (2010). Classical Music for Rock Fans?: Novel Recommendations for Expanding User Interests. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010). Toronto, Canada, October 2010, pp. 949-958.

M. E. J. Newman (2010). Networks, an Introduction, 1st Edition. Oxford University Press, 2010.

X. Ning, C. Desrosiers and G. Karypis (2015). A comprehensive survey of neighborhood-based recommendation methods. In Recommender Systems Handbook, F. Ricci, L. Rokach, and B. Shapira (Eds.). Springer, 2015, pp. 37-76.

X. Ning and G. Karypis (2011). SLIM: Sparse Linear Methods for Top-N Recommender Systems. In Proceedings of the IEEE 11th International Conference on Data Mining (ICDM 2011). Vancouver, Canada, December 2011, pp 497-506

J. Oh, S. Park, H. Yu, M. Song and S. T. Park (2011). Novel Recommendation Based on Personal Popularity Tendency. In Proceedings of the IEEE 11th International Conference on Data Mining (ICDM 2011). Vancouver, Canada, December 2011, pp. 507-516.

K. Onuma, H. Tong and C. Faloutsos (2009). TANGENT: A Novel, "Surprise-me", Recommendation Algorithm. In Proceedings of the 15th ACM International Conference on Knowledge Discovery and Data Mining (KDD 2009). Paris, France, June 2009, pp. 657-666.

E. Pariser (2012). The Filter Bubble: How the New Personalized Web Is Changing What We Read and How We Think. Penguin Books, London, UK, 2012.

B. Pradel, N. Usunier and P. Gallinari (2012). Ranking with non-random missing ratings: influence of popularity and positivity on evaluation metrics. In Proceedings of the 6th ACM Conference on Recommender Systems (RecSys 2012). Dublin, Ireland, September 2012, pp. 147-154.

J. Ratkiewicz, S. Fortunato, A. Flammini, F. Menczer and A. Vespignani (2010). Characterizing and Modeling the Dynamics of Online Popularity. Physical Review Letters 105(15). APS, Ridge, NY, USA, October 2010.

S. Rendle, C. Freudenthaler, Z. Gantner and L. Schmidt-Thieme (2009). BPR: Bayesian personalized ranking from implicit feedback. In Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI 2009). Montreal, Quebec, Canada, June 2009, pp. 452-461 .

F. Ricci, L. Rokach and B. Shapira (2015) (Eds.). Recommender Systems Handbook, 2nd Edition. Springer, 2015.

S. E. Robertson (1977). The Probability Ranking in IR. Journal of Documentation 33(4). January 1977, pp. 294-304.

M. J. Salganik, P. S. Dodds and D. J. Watts (2006). Experimental Study of Inequality and Unpredictability in an Artificial Cultural Market. Science 311(5762), February 2006, pp. 854-856.

A. Said and A. Bellogín (2014). Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks. In Proceedings of the 8th ACM Conference on Recommender Systems (RecSys 2014). San Jose, CA, USA, October 2014, pp. 129-136.

T. Schnabel, A. Swaminathan, A. Singh, N. Chandak, and T. Joachims (2016). Recommendations as Treatments: Debiasing Learning and Evaluation. In Proceedings of the 33rd International Conference on Machine Learning (ICML 2016). NewYork, NY, USA, June 2016, pp. 1670-1679.

G. Shani and A. Gunawardana (2015). Evaluating recommendation systems. In Recommender Systems Handbook (2nd edition), F. Ricci, L. Rokach, and B. Shapira (Eds.). Springer, 2015, pp. 265-308.

A. Sharma, J. M. Hofman and D. J. Watts (2015). Estimating the Causal Impact of Recommendation Systems from Observational Data. In Proceedings of the 16th ACM Conference on Economics and Computation (EC 2015). Portlan, Oregon, June 2015, pp. 453-470.

H. Shen, D. Wang, C. Song and A. L Barabási (2014). Modeling and Predicting Popularity Dynamics via Reinforced Poisson Processes. In Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI 2014). Hilton, Québec, July 2014, pp. 291-297.

A. Sinha, D. F. Gleich, and K. Ramani. 2016. Deconvolving Feedback Loops in Recommender Systems. In Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016). Barcelona, Spain, December 2016, pp. 3243-3251.

D. Siroker and P. Koomen (2015). A/B testing: the most powerful way to turn clicks into customers. John Wiley & Sons Inc, Hoboken, New Jersey, USA, 2015.

B. Smyth and P. McClave (2001). Similarity vs. diversity. In Proceedings of the 4th International Conference on Case-Based Reasoning (ICCBR 2001). Springer-Verlag, London, UK, pp. 347-361.

H. Steck (2013). Evaluation of recommendations: rating-prediction and ranking. In Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (RecSys 2013). Hong Kong, October 2013, pp. 213-220.

H. Steck (2011). Item popularity and recommendation accuracy. In Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (RecSys 2011). Chicago, IL, USA, October 2011, pp. 125-132.

H. Steck (2010). Training and testing of recommender systems on data missing not at random. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2010). Washington, DC, USA, July 2010, pp. 713-722.

J. Su, A. Sharma and S. Goel (2016). The Effect of Recommendations on Network Structure. In Proceedings of the 25th International Conference on World Wide Web (WWW 2016). Montréal, Canada, April 2016, pp. 1157-1167.

A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni (2017). Off-policy Evaluation for Slate Recommendation. In Proceedings of Advances in Neural Information Processing Systems *30* (NIPS 2017). Curran Associates, Inc., Red Hook, NY, USA, pp. 3632-3642.

G. Szabo and B. A. Huberman (2010). Predicting the Popularity of Online Content. Communications of the ACM 53(8) ACM, New York, NY, August 2010, pp. 80-88.

S. Vargas and P. Castells (2014). Improving sales diversity by recommending users to items. In Proceedings of the 8th ACM Conference on Recommender Systems (RecSys 2014). Foster City, Silicon Valley, USA, October 2014, pp. 145-152.

L. Yang, Y. Cui, Y. Xuan, C. Wang, S. Belongie, and D. Estrin (2018). Unbiased Offline Recommender Evaluation for Missing-Not-At-Random Implicit Feedback. In Proceedings of the 12th ACM Conference on Recommender Systems (RecSys 2018). Vancouver, Canada, October 2017, pp. 279-287.

T. Wang and D. Wang (2014). Why Amazon's Ratings Might Mislead You: The Story of Herding Effects. Big Data 2(4), December 2014, pp.196-204.

Q. Wu, H. Wang, L. Hong and Y. Shi (2017). Returning is Believing: Optimizing Long-term User Engagement in Recommender Systems. In Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (CIKM 2017). Singapore, Singapore, November 2017, pp. 1927-1936.

C. Zhai and J. D. Lafferty (2004). A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems 22 (2). ACM, New York, NY, USA, April 2004, pp. 179-14.

P. Zhang, M. Li, L. Gao, Y. Fan and Z. Di (2014). Characterizing and Modeling the Dynamics of Activity and Popularity. PLOS ONE 9(2). Public Library of Science, San Francisco, CA, USA, February 2014.

X. Zhao, Z. Niu and W. Chen (2013). Opinion-Based Collaborative Filtering to Solve Popularity Bias in Recommender Systems. In Proceedings of the 24th International Conference on Database and Expert Systems Applications (DEXA 2013). Prague, Czech Republic, August 2013, pp. 426-433.

M. Zhang and N. Hurley (2008). Avoiding monotony: Improving the diversity of recommendation lists. In Proceedings of the 2nd ACM Conference on Recommender Systems (RecSys 2008). Lousanne, Switzerland, October 2008, pp. 123-130.

C. N. Ziegler, S.M. McNee, J.A. Konstan and G. Lausen (2005). Improving recommendation lists through topic diversification. In Proceedings of the 14th International Conference on World Wide Web (WWW 2005). Chiba, Japan, May 2005, pp. 22-32.